# THE NUMERICAL SOLUTION AND ANALYSIS OF AIRPLANE SPIN EQUATIONS MODELED IN A FIXED COORDINATE SYSTEM

Robert Louis Champoux

Library
Naval Postgraduate School
Monterey, California 93940

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

# THESIS

The Numerical Solution and Analysis
of Airplane Spin Equations
Modeled in a Fixed Coordinate System

by

Robert Louis Champoux

Thesis Advisor: Louis V. Schmidt

December 1972

T153081

Approved for public release; distribution unlimited.

# The Numerical Solution and Analysis of Airplane Spin Equations Modeled in a Fixed Coordinate System

by

Robert Louis Champoux
Lieutenant, United States Navy
B.S., United States Naval Academy, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1972

Library
Naval Postgraduate School
Monterey, California 93940

## ABSTRACT

Three forms of the airplane spin equations of motion, derived by Buehler in Reference [1], form the basis for the development of a computer program designed to seek dynamically stable equilibrium solutions of a spinning aircraft. The program incorporates two solution techniques: one based upon Euler integration, the other, a version of minimization by gradient search. Secondary programs are developed to (1) generate power-off glide parameters for use in the validation of the equations of motion, and (2) evaluate equation residuals obtained from a grid of initial conditions over the potential solution space. F-111 and F-4 aerodynamic force and moment models were utilized to evaluate the solution methods and equations of motion. The numerical results indicate that the F-111 and F-4 data are not representative of the actual aircraft and, therefore, it is highly unlikely that dynamically stable equilibrium solutions can be achieved from these models. The utility of the two solution methods is evaluated and the numerical results are analyzed in order to gain insight into the optimal application of the three forms of the equations of motion. The paper concludes with a discussion concerning the qualitative validation of the equations of motion.

TABLE OF CONTENTS


I. INTRODUCTION . . . . . 11

A. BACKGROUND . . . . . 11

B. GENERAL . . . . . 12

C. SCOPE . . . . . 12

II. ANALYTIC CONSIDERATIONS . . . . . 14

A. COORDINATE SYSTEM . . . . . 14

B. EULER ORIENTATION ANGLES . . . . . 16

C. THE EQUATIONS OF MOTION . . . . . 18

D. ANCILLIARY RELATIONSHIPS AND DEPENDENT VARIABLES . . . . . 19

E. AERODYNAMIC FORCE AND MOMENT COEFFICIENTS . . . . . 21

F. SOLVING THE EQUATIONS OF MOTION . . . . . 23

G. THE INTEGRATION SOLUTION METHOD . . . . . 23

H. THE GRADIENT SOLUTION METHOD . . . . . 27

III. AIRCRAFT SPIN SOLUTION COMPUTER PROGRAM . . . . 31

A. GENERAL . . . . . 31

B. PROGRAM USAGE . . . . . 31

C. INPUT DATA . . . . . 31

D. OUTPUT DATA . . . . . 33

E. PROGRAM MODULE DESCRIPTION . . . . . 35

IV. AIRPLANE SPIN EQUATION VALIDATION . . . . . 38

V. NUMERICAL ANALYSIS AND CONCLUSIONS . . . . . 42

A. AERODYNAMIC FORCE AND MOMENT MODELS . . . 42

B. SOLUTION METHOD EVALUATION . . . . . 42

1. Integration Method . . . . . 42

2. Gradient Method . . . . . 48

C. ANALYSIS OF THE F-111 AND F-4 STEADY-STATE SOLUTION SPACES . . . . . 50

VI. SUMMARY OF THE MAJOR CONCLUSIONS AND THE RECOMMENDATIONS . . . . . 55

APPENDIX A - The Aircraft Equations of Motion . . . . . 56

APPENDIX B - The Aircraft Spin Solution Computer Program . . 62

APPENDIX C - Criterion Function Search Computer Program . . 111

APPENDIX D - Equation Validation Computer Program (TRIM) . . . . . 121

APPENDIX E - Aerodynamic Force and Moment Model Data . . . 125

BIBLIOGRAPHY . . . . . 137

INITIAL DISTRIBUTION LIST . . . . . 138

FORM DD 1473 . . . . . 139

## LIST OF TABLES


| Table | | Page |
|---|---|---|
| I | Direction Cosines . . . . . | 16 |
| II | Aerodynamic Force and Moment Derivatives . . . . . | 22 |
| III | Pertinent Data Relating to Equation Validation Using TRIM and the Spin Solution Computer Program . . . . . | 40 |
| IV | Initial Conditions for Equation Validation . | 41 |
| V | Results from a Typical Integration Solution Effort . . . . . | 47 |
| VI | Summary of an F-111 Gradient Solution Effort . . . . . | 49 |
| VII | Initial Conditions Generated by the Criterion Function Search Program . . . | 51 |
| VIII | Gradient and Integration Results from Criterion Search Generated Initial Conditions . . . . . | 53 |
| B-1 | Force and Moment Coefficients . . . . . | 64 |
| B-2 | NSFLAG Integer Assignments . . . . . | 66 |
| C-1 | Pertinent Data Relating to the Criterion Function Search Program . . . . | 113 |
| D-1 | TRIM Output Data . . . . . | 122 |

## LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 1 | The Coordinate System Used to Model the Spin | 15 |
| 2 | A General Schematic of Euler Angle Relationships | 15 |
| 3 | Definition of Positive $\alpha$ and $\beta$ Angles | 20 |
| 4 | A Schematic Representation of the Integration Solution Method | 26 |
| 5 | A Schematic Representation of the Gradient Solution Method | 30 |
| 6 | Airplane Spin Solution Program Hierarchy | 32 |
| 7 | Explanation of the Airplane Spin Solution Printed Output | 34 |
| 8 | Schematic of Aerodynamic Parameters for an Airplane in a Power-off Glide | 39 |
| 9 | Schematic of a Typical Integration Solution Effort | 46 |
| 10 | Criterion Function Search Procedures and Results | 52 |
| B-1 | Data Deck Composition for the Airplane Spin Solution Program | 63 |
| B-2 | MAIN Program Flowchart | 69 |
| B-3 | SOLVIT Subroutine Flowchart | 70 |
| B-4 | GAWAIN Subroutine Flowchart | 71 |
| B-5 | EQNS Subroutine Flowchart | 72 |
| B-6 | EULER Subroutine Flowchart | 73 |
| B-7 | ANGLE Subroutine Flowchart | 74 |
| B-8 | INTER Subroutine Flowchart | 75 |
| B-9 | FIX Subroutine Flowchart | 76 |

Figure Page

C-1 Abbreviated Flowchart for the Criterion Search Computer Program . . . . 112

## TABLE OF SYMBOLS

The definitions of the symbols used in this paper are as follows:

| | |
|---|---|
| $C_g$ | Center of gravity |
| $\bar{c}$ | Mean aerodynamic chord, feet |
| g | Acceleration of gravity, ft/sec$^2$ |
| $I_x, I_y, I_z$ | Airplane principal moments of inertia, slug-ft$^2$ |
| M | Airplane mass, slugs |
| N | Load factor; number of "g's" |
| R | Spin Radius, feet |
| S | Criterion function value; equal to the sum of the absolute values of the equation residuals, SUM |
| s | Airplane wing area, ft$^2$ |
| t | Time, seconds |
| $\Delta t$ | Increment of time, seconds |
| $V_{cg}$ | Airplane cg velocity, ft/sec |
| X, Y, Z | Airplane principal axis system |
| $X_1, Y_1, Z_1$ | Reference Cartesian axis system |
| $Z_o$ | Airplane altitude, feet |
| $\dot{Z}_o$ | Rate of ascent, ft/sec |
| $\alpha$ | Airplane angle of attack, alpha |
| $\alpha_{ij}$ | Direction cosines in terms of Euler angles (Table I) |
| $\beta$ | Airplane sideslip angle, Beta |
| $\dot{\vartheta}$ | Spin rate, per second |
| $\delta_a, \delta_r, \delta_e$ | Aileron, rudder and elevator deflections, degrees |
| $\Theta, \Psi, \Phi$ | Airplane principal axis system orientation Euler angles with respect to the $X_1$, $Y_1$, $Z_1$ system |

| | |
|---|---|
| $\Theta_p, \psi_r, \phi_y$ | Airplane orientation in pitch, roll and yaw with respect to the $X_1$, $Y_1$, $Z_1$ system |
| $\rho$ | Air density, slugs/ft$^3$ |
| W | Aircraft weight, lbs |

## FOREWORD AND ACKNOWLEDGMENTS

The purpose of this effort was to develop a computer program which would numerically predict the dynamically stable spin characteristics of an aircraft. Essential to the task was the analysis and validation of the airplane spin equations derived by LCDR R. E. Buehler and described in Reference [1]. Although the final results were inconclusive, if any success is ultimately realized from the program it will be due in a large part to LCDR Buehler who, during the concurrent development, withstood many frontal assaults upon the veracity of his equations. He many times patiently rederived various aspects of his work only to learn that the intricacies of the computer program were the underlying culprits. Appreciation is also expressed to Professor L. V. Schmidt, who significantly aided in the validation of the equations and for providing constructive criticisms concerning manuscript preparation. The author is also indebted to Professors Gawain and Redlin of the NPS faculty whose council helped in the utilization of the gradient method and criterion search aspects of the analysis. Finally, the staff of the William Randolph Church Computer Center is to be commended for the consultative support which they provided.

# I. INTRODUCTION

## A. BACKGROUND

Aircraft flight tests frequently reveal significant differences between the predicted and the actual spin characteristics of high performance systems. Such differences often result in expensive engineering changes to the aircraft design and tend to reduce the mission capability of the aircraft. Furthermore, if the final spin characteristics are such that pilots are not permitted to practice the recovery from intentional spins, then many will consider the outer regions of the performance envelope with trepidation resulting in further degradation of the weapon system's effectiveness. The need therefore, to accurately predict the spin characteristics of proposed aircraft designs is of paramount importance.

Historically, efforts towards spin performance prediction have been concentrated in two areas: the development of system time histories and the analysis of free-spinning model test results. System time histories involve the simultaneous solution of the aircraft equations of motion with the required force and moment coefficients being determined from wind tunnel model data. Solutions are obtained using digital computer programs which numerically integrate the appropriate equations of motion. There has been minimal success towards achieving a time history method that will accurately predict spin performance due to the critical dependence upon entry conditions. Free-spinning model tests utilize instrumented dynamically scaled models which are generally deployed into vertical wind tunnels. This method however has a limited capability in simulating the full scale

spin environment (Reynolds, Mach, and Froude Numbers and the spin entry mechanism) and often the tests fail to identify all of the possible spin modes that may be experienced by the full scale vehicle.

## B. GENERAL

This paper represents the final report on the development of a digital computer program designed to analytically predict the airplane spin performance using a unique set of aircraft equations of motion and simplifying assumptions regarding the nature of spin mode solutions. The primary hypothesis upon which the development of the computer program was predicated was the unique dynamically stable equilibrium (steady-state) solution for each aircraft spin mode. This assumption implies that all motion will ultimately be damped to a steady-state solution. Using the above hypothesis, the dynamically stable equilibrium solution may be determined directly without the encumbrances of a time history approach, with its required dependence upon entry conditions. The hypothesis ignores, however, the possible (and perhaps most probable) existence of a dynamically unstable equilibrium solution which could also yield a stable trajectory in 'state space'.

## C. SCOPE

A rigorous research effort into the development of a means of predicting airplane spin modes would involve the following stages:

I. Literature search

II. Derivation of the mathematical model

III. Verification of the mathematical model by using time history comparisons with baseline data

IV. Development of a computer program designed to seek spin solutions

V. Verification of the computer program by utilizing wind tunnel data modified for simulator use

VI. Determine and/or predict "real world" spin modes.

Stages I and II are described in References [7] and [1]; while the base line data required for phase III is included in Reference [2]. The purpose of this effort was an attempt to accomplish steps IV and VI. This rather large undertaking was conducted concurrently with LCDR Buehler, who developed the applicable equations and coordinate systems required by stage II. The limited time available precluded a numerical validation of the computer program (phase III).

## II. ANALYTIC CONSIDERATIONS

### A. COORDINATE SYSTEM

The coordinate system used in the mathematical model is a combination of the airplane principal axis system and a fixed cylindrical system. The vertical (z) axis of the fixed cylindrical system represents the initial central axis of motion. The cylindrical system locates the aircraft center of gravity in terms of the altitude coordinate ($z_o$), the spin radius coordinate (R) and the angular position coordinate ($\gamma$). The orientation of the aircraft with respect to the cylindrical system is described in terms of Euler angles. A cartesian coordinate system is fixed at the cg position on the R vector with its axis ($x_1$) in the (-R) direction and the ($z_1$) axis is oriented in the (-z) direction. The cartesian system provides a reference for the orientation Euler angles whereby zero values of ($\theta$), ($\psi$), and ($\phi$) would yield an upright, wings level airplane with its principal (x) axis pointed inward in the (-R) direction. Figure 1 depicts the details of the coordinate system; the positive directions of the position coordinates are also shown. As Buehler noted "The rationale behind this choice of coordinate system is that it more simply (in a mathematical sense) represents the motion being modeled." [Reference 1]. The advantage of this hybrid coordinate system is that it obviates a steady-state spin solution. The motion of the spinning aircraft is easily recognized by constant values of the six independent coordinate variables (R, $\dot{z}_o$, $\dot{\gamma}$, $\theta, \psi, \phi$). Additionally the complexities of the stall, departure, and incipient spin motion have no impact on the results since the objective is to determine the steady-state spin modes.

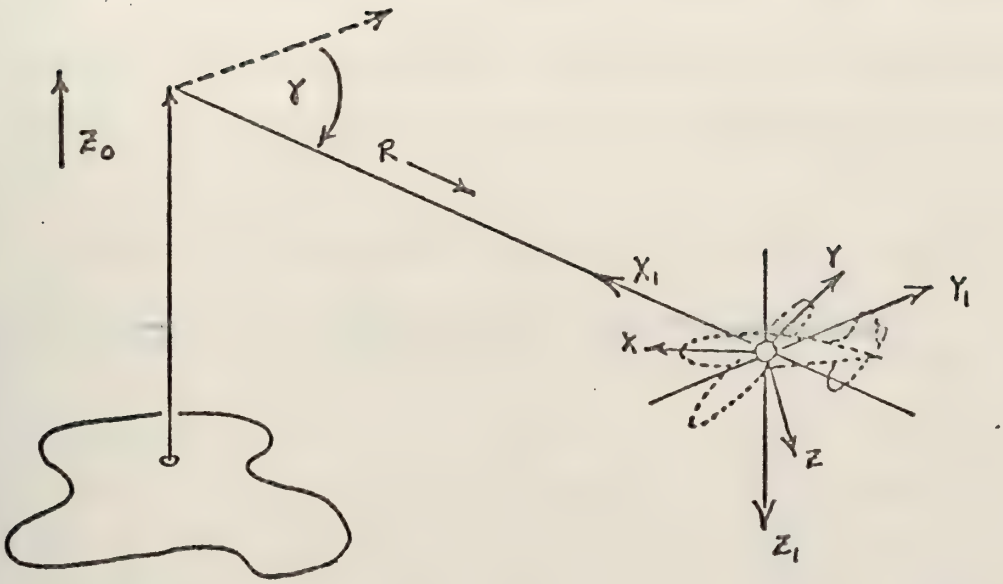



Fig. 1. The Coordinate System Used to Model the Spin

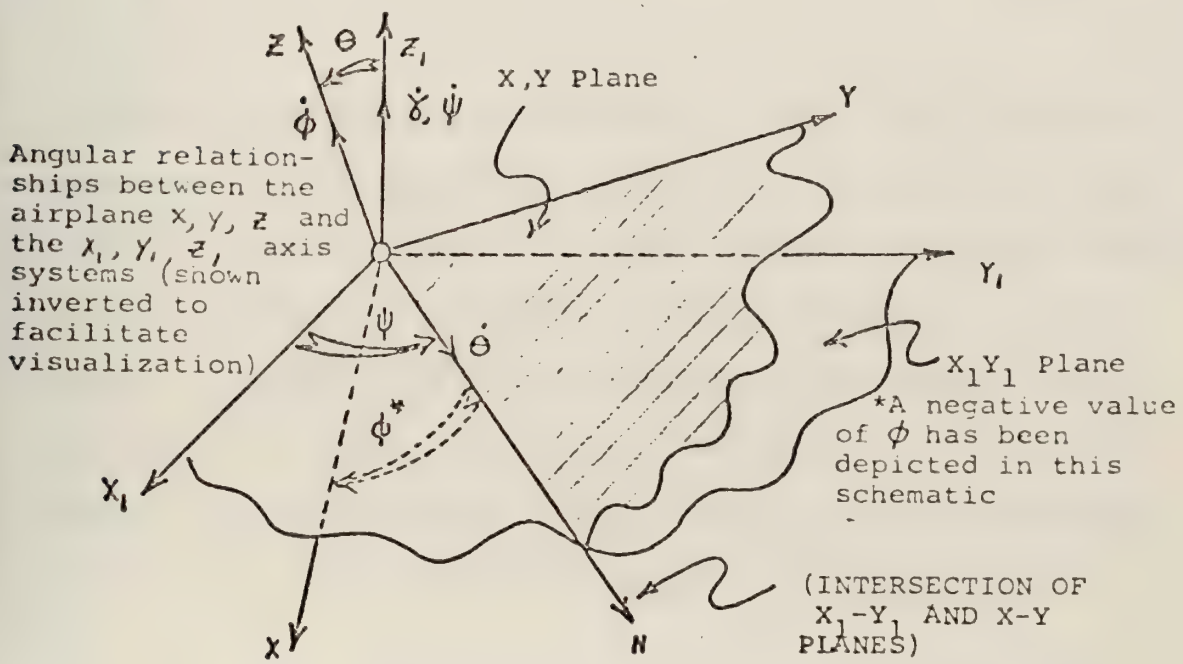



Fig. 2. A General Schematic of Euler Angle Relationships

B. EULER ORIENTATION ANGLES

The Euler Angle relationships utilized in the mathematical model involve rotations which are measured relative to the vertical ($z_1$) axis and to the intersection of the (x, y) plane and the ($x_1$, $y_1$) plane. This system yields equations for the direction cosines as listed in Table I.

Table I: Direction Cosines in Terms of Euler Angles

| Cosines of Angles between $X, Y, Z$ and $X_1, Y_1, Z_1$ | | | |
|---|---|---|---|
| | $X$ | $Y$ | $Z$ |
| $X_1$ | $a_{11} = \cos\phi\cos\psi - \sin\phi\sin\psi\cos\theta$ | $a_{21} = -\sin\phi\cos\psi - \cos\phi\sin\psi\cos\theta$ | $a_{31} = \sin\theta\sin\psi$ |
| $Y_1$ | $a_{12} = \cos\phi\sin\psi + \sin\phi\cos\psi\cos\theta$ | $a_{22} = -\sin\phi\sin\psi + \cos\phi\cos\psi\cos\theta$ | $a_{32} = -\sin\theta\cos\psi$ |
| $Z_1$ | $a_{13} = \sin\theta\sin\phi$ | $a_{23} = \sin\theta\cos\phi$ | $a_{33} = \cos\theta$ |

Inherent in this orientation system, however, is the difficulty in visualizing the final orientation without employing sketches or digital graphics equipment. This can be readily appreciated by referring to the general schematic of the Euler angle relationships depicted in Figure 2. In order to alleviate the visualization problem, Buehler, in Reference [1], derived relationships between a set of orientation angles with a sequence of pitch ($\Theta_p$), roll ($\Psi_r$), and then yaw ($\Phi_y$) and the Euler angles depicted in Figure 2.[1] The equations for

[1] Hereafter, and in the computer program, the orientation angles will be referred to as the 'ordered' Euler angles, while the orientation angles depicted in Figure 2 will be referred to as the 'reference' Euler angles.

converting the ordered Euler angles to the reference Euler angles are:

$$\Phi = \Psi_y - TAN^{-1}\left\{TAN\left(\frac{\Theta_p}{2}\right) TAN\left(\frac{\Psi_R}{2}\right)\right\} - TAN^{-1}\left\{\frac{TAN\left(\frac{\Theta_p}{2}\right)}{TAN\left(\frac{\Phi_R}{2}\right)}\right\} \quad (1)$$

$$\Psi = TAN^{-1}\left\{\frac{TAN\frac{\Theta_p}{2}}{TAN\frac{\Phi_R}{2}}\right\} - TAN^{-1}\left\{TAN\left(\frac{\Theta_p}{2}\right) TAN\left(\frac{\Phi_R}{2}\right)\right\} \quad (2)$$

$$\Theta = 2\,TAN^{-1}\left\{\frac{SIN\left[TAN^{-1}\left\{TAN\left(\frac{\Theta_p}{2}\right) TAN\left(\frac{\Phi_r}{2}\right)\right\}\right]}{SIN\left[TAN^{-1}\left\{\frac{TAN\left(\frac{\Theta_p}{2}\right)}{TAN\left(\frac{\Psi_r}{2}\right)}\right\}\right] TAN\left(\frac{\Phi_r}{2}\right)}\right\} \quad (3)$$

The equations for obtaining ($\Theta_p$), ($\Psi_r$) and ($\Phi_y$) from the reference Euler angles are:

$$\Psi_y = \Phi + TAN^{-1}\left\{\frac{TAN\left(\frac{\Psi}{2}\right) SIN\left(\frac{\pi/2 - \Theta}{2}\right)}{SIN\left(\frac{\pi/2 + \Theta}{2}\right)}\right\} + TAN^{-1}\left\{\frac{TAN\left(\frac{\Psi}{2}\right) SIN\left(\frac{\pi/2 - \Theta}{2}\right)}{Cos\,\frac{\pi/2 + \Theta}{2}}\right\} \quad (4)$$

$$\Theta_p = TAN^{-1}\left\{\frac{TAN\left(\frac{\Psi}{2}\right) Cos\left(\frac{\pi/2 - \Theta}{2}\right)}{Cos\left(\frac{\pi/2 + \Theta}{2}\right)}\right\} - TAN^{-1}\left\{\frac{TAN\left(\frac{\Psi}{2}\right) SIN\left(\frac{\pi/2 - \Theta}{2}\right)}{Cos\left(\frac{\pi/2 + \Theta}{2}\right)}\right\} \quad (5)$$

$$\Phi_R = \frac{\pi}{2} - 2\,TAN^{-1}\;\frac{SIN\left[TAN^{-1}\left\{\dfrac{TAN\left(\frac{\gamma}{2}\right)SIN\left(\frac{\pi/2-\theta}{2}\right)}{SIN\left(\frac{\pi/2+\theta}{2}\right)}\right\}\right]}{SIN\left[TAN^{-1}\left\{\dfrac{TAN\left(\frac{\gamma}{2}\right)COS\left(\frac{\pi/2-\theta}{2}\right)}{COS\left(\frac{\pi/2+\theta}{2}\right)}\right\}\right]TAN\left(\frac{\pi/2-\theta}{2}\right)} \qquad (6)$$

The above equations have been numerically verified and validated for quadrants I and IV ($\pm$ 90°). This is sufficient for an aircraft oriented in an upright spin, however, for the case of an inverted spin where the aircraft will encounter quadrants II and III, the above relationships are invalid.

## C. THE EQUATIONS OF MOTION

The spin is an uncontrolled large angle, six degree of freedom motion experienced by an aircraft in the stalled aerodynamic region. Since the spinning phenomenon is a complex motion which is influenced by a host of nonlinear variables, the equations which describe that motion are both nonlinear and coupled. Buehler, in Reference [1], derived the aircraft spin equations of motion utilized in the computer program. The equations and modifications thereto are described below.

### 1. The Full Equations

The equations of motion were cast in a more useable form by solving each equation for the corresponding acceleration parameter. This form avoids obscuration of the physical nature of the problem and places the numerical analyst in a better position to monitor, trouble-shoot and interpret iterative solution results.

2. The Modified Equations

The full equations were further modified by recognizing that only dynamically stable (or quasi-stable) equilibrium solutions were sought. Thus angular orientation rates ($\dot{\Theta}, \dot{\psi}, \dot{\Phi}$) were equated to zero, which in turn eliminated a considerable number of terms. The R parameter was retained because it effects the values of $\alpha$, $\beta$, and $V_r$ to the extent that when ignored, it seriously impeded the iterative solution effort.

3. The Short Equations

The modified equations were further simplified by equating the R parameter to zero. This form is useful when seeking purely steady state solutions using the more mathematically elegant solution methods such as the gradient search method described in Section H.

The full, modified, and short equations are listed in Appendix I.

D. ANCILLIARY RELATIONSHIPS AND DEPENDENT VARIABLES

In order to utilize conventional aerodynamic force and moment coefficients in the equations of motion, certain ancilliary relationships were needed. As derived by Buehler, and as depicted in Figure 3:

Angle of Attack:

$$\alpha = TAN^{-1}\left\{\frac{\alpha_{31}\dot{R} + \alpha_{32}\dot{\gamma}R + \alpha_{33}\dot{Z}_o}{\alpha_{11}\dot{R} + \alpha_{12}\dot{\gamma}R + \alpha_{13}\dot{Z}_o}\right\} \tag{7}$$

Sideslip Angle:

$$\beta = TAN^{-1}\left\{\frac{\alpha_{21}\dot{R} + \alpha_{22}\dot{\gamma}R + \alpha_{23}\dot{Z}_o}{\alpha_{11}\dot{R} + \alpha_{12}\dot{\gamma}R + \alpha_{13}\dot{Z}_o}\right\} \tag{8}$$

Relative Wind Velocity:

$$V_R = \left\{ \dot{R}^2 + (\dot{\gamma} R)^2 + \dot{Z}_o^2 \right\}^{1/2} \tag{9}$$

Roll Rate:

$$p = (\dot{\psi} + \dot{\gamma}) \alpha_{13} + \dot{\Theta} \cos \Phi \tag{10}$$

Pitch Rate:

$$q = (\dot{\psi} + \dot{\gamma}) \alpha_{23} - \dot{\Theta} \sin \Phi \tag{11}$$

Yaw Rate:

$$r = (\dot{\psi} + \dot{\gamma}) \alpha_{33} + \dot{\Phi} \tag{12}$$

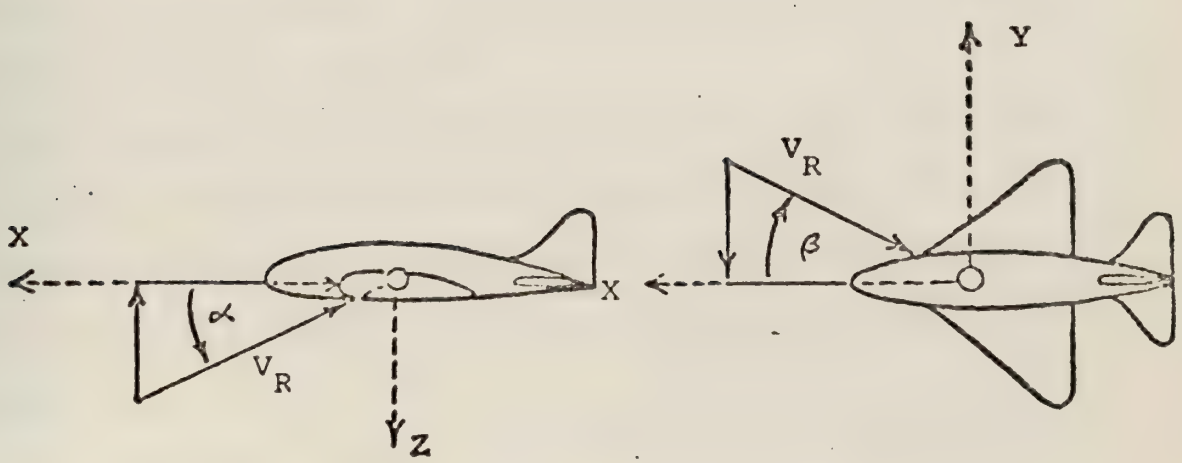



Figure 3

Definition of Positive $\alpha$ and $\beta$ Angles

## E. AERODYNAMIC FORCE AND MOMENT COEFFICIENTS

The general form of the aerodynamic force and moment coefficients employed in the equations of motion are as follows:

$$C_{F_i} = C_{F_{i_1}} + C_{F_{i_2}} f(p) + C_{F_{i_3}} f(q) + C_{F_{i_4}} f(r) + C_{F_{i_5}} f(\delta a) + C_{F_{i_6}} f(\delta r) + C_{F_{i_7}} f(\delta e) \qquad i = x, y, z \tag{13}$$

$$C_{m_i} = C_{m_{i_1}} + C_{m_{i_2}} f(p) + C_{m_{i_3}} f(q) + C_{m_{i_4}} f(r) + C_{m_{i_5}} f(\delta a) + C_{m_{i_6}} f(\delta r) + C_{m_{i_7}} f(\delta e) \qquad i = x, y, z \tag{14}$$

where $\delta a$ $\delta r$ $\delta e$ represent the angular deflections of the ailerons, rubber and elevator respectively.

All of the above coefficients on the right side of the equality sign are aerodynamic force and moment derivatives as defined in Table II. They must be experimentally determined and tabulated for a wide range of $\alpha$ and $\beta$ vlaues. f(p, q, or r) are functions of roll, pitch and yaw rates.

The coefficients in Table II utilize $\bar{c}$ as a common characteristic length, thus allowing the use of a single length parameter input to the computer program. Since $\dot{\alpha}$ and $\dot{\beta}$ are zero for the steady spin, aerodynamic derivatives with respect to these two parameters have not been included in Table II.

| | SUBSCRIPT j | | | | | | |
|---|---|---|---|---|---|---|---|
| i=x, y, or z | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $C_{Fij}$ | $C_{Fi}(\alpha,\beta)$ | $\frac{\partial C_{Fi}}{\partial(\frac{p\bar{c}}{2V})}$ | $\frac{\partial C_{Fi}}{\partial(\frac{q\bar{c}}{2V})}$ | $\frac{\partial C_{Fi}}{\partial(\frac{r\bar{c}}{2V})}$ | $\frac{\partial C_{Fi}}{\partial \delta_a}$ | $\frac{\partial C_{Fi}}{\partial \delta_r}$ | $\frac{\partial C_{Fi}}{\partial \delta_h}$ |
| $C_{Mij}$ | $C_{Mi}(\alpha,\beta)$ | $\frac{\partial C_{Mi}}{\partial(\frac{p\bar{c}}{2V})}$ | $\frac{\partial C_{Mi}}{\partial(\frac{q\bar{c}}{2V})}$ | $\frac{\partial C_{Mi}}{\partial(\frac{r\bar{c}}{2V})}$ | $\frac{\partial C_{Mi}}{\partial \delta_a}$ | $\frac{\partial C_{Mi}}{\partial \delta_r}$ | $\frac{\partial C_{Mi}}{\partial \delta_h}$ |

Table II

Definition of the Aerodynamic Derivatives[2]

The validity of any spin solution results is dependent upon how well the aerodynamic forces and moments have been modeled. One needs only to visualize the different dynamic environment that exists between the static model in a wind tunnel and the actual airplane experiencing three dimensional stalled flow in a spin. Herein lies the greatest weakness in the analytic spin solution effort. The static wind tunnel data utilized to determine the above 42 coefficients provides, at best, a very crude model of the actual conditions encountered by a spinning aircraft. However, for want of better data, the conventional data was utilized in order to exercise the program and to seek a first-order approximation to a spin solution.

---

[2] Aerodynamic derivatives with respect to $\dot{\alpha}$ and $\dot{\beta}$ have not been included since $\dot{\alpha}$ and $\dot{\beta}$ are zero for the steady spin.

## F. SOLVING THE EQUATIONS OF MOTION

A casual examination of the equations of motion found in Appendix II reveals that they have the following characteristics:

1. ordinary second order differential equations
2. extensively coupled
3. highly nonlinear
4. force and moment terms are a function of three-dimensional flow over the geometry of the airplane.

The above characteristics preclude direct analytical integration and thus one is forced to seek a numerical solution scheme which will provide the greatest insight into the physical problem and also provide some degree of uniform convergence. Additionally, from numerical considerations the extreme length of the equations dictate the need for a method which maintains numerical significance and does not require excessive computational time.

Two iterative solution methods were selected; the integration method which was utilized for steady and quasi-steady state solutions and the gradient method for the purely steady-state solution efforts. In both instances, the set of equations could be visualized as a "black box" function transform with independent variables as inputs and the equation residuals as outputs.

## G. THE INTEGRATION SOLUTION METHOD

The integration method can be visualized as the incremental tracking of the trajectory of a multiple degree of freedom spring-mass-damper system. The incremental changes are generated by "psuedo" integration of the aerodynamic force and moment inbalances manifested in the equation residuals. This set of incremental changes constitutes

the "trajectory". The psuedo integration is accomplished by taking advantage of the form of the equations whereby the residuals are essentially the accelerations of the six spin degrees of freedom. Application of a common increment of time to elementary Euler integration formulae allows the computation of a dynamically consistent set of incremental changes in the independent variables and thus generates new values for the subsequent iteration. Essentially, the method duplicates, in a numerical sense, what the aircraft experiences under actual flight conditions.

One of the major advantages of casting the equations of motion in a cylindrical reference frame, is the ease with which tumbling motion can be separated from the motion associated with a steady state spin. Constant values of $Z_0$, R, $\dot{\delta}$, $\Theta$, $\Psi$, and $\Phi$ indicate a steady state solution, while non-zero values of $\dot{\Theta}$, $\dot{\Psi}$, or indicate tumbling motion. The computational scheme discards any angular momentum inconsistent with a steady-state spin and thus on each iteration any tumbling motion is suppressed. Such motion, if retained, could possibly carry the solution path (trajectory) through a steady state spin solution.

The iterative procedure used in the computer was basically very simple. In response to a set of initial conditions ($\dot{Z}_0$, R, $\dot{\delta}$, $\Theta$, $\Psi$, $\Phi$) and related constants, the corresponding accelerations were computed. Application of a common time increment ($\Delta t$) to the basic Euler integration formulae yields a set of incremental changes which are then added to the previous set of independent variables to obtain a new set of initial conditions. The iteration process continues until the sum of the absolute values of the accelerations, denoted as S, is reduced

to less than some specified value. The "spring-mass-damper" system thus moves closer to equilibrium as it converges to a steady state solution.

The acceleration $\ddot{Z}_0$ and $\ddot{R}$ can be computed directly from the independent and dependent variables. The other accelerations ($\ddot{\gamma}$, $\ddot{\theta}$, $\ddot{\psi}$, $\ddot{\Phi}$) however, must be computed by a looping routine since the associated equations are coupled to all of the angular accelerations. The looping ceases once a consistent set of accelerations is obtained.

The incremental changes are obtained using the following equations:

$$\dot{Z}_{o(N+1)} = \dot{Z}_{o(N)} + (\ddot{Z}_o \Delta t)a \tag{15}$$

$$\dot{R}_{(N+1)} = \dot{R}_{(N)} + (\ddot{R} \Delta t)b \tag{16}$$

$$R_{(N+1)} = R_{(N)} + (\dot{R}_{(N+1)} \Delta t + \ddot{R} \frac{\Delta t^2}{2})b \tag{17}$$

$$\dot{\gamma}_{(N+1)} = \dot{\gamma}_{(N)} + (\ddot{\gamma} \Delta t)c \tag{18}$$

$$\dot{\theta}_{(N+1)} = \dot{\theta}_{(N)} + (\ddot{\theta} \Delta t)d \tag{19}$$

$$\theta_{(N+1)} = \theta_{(N)} + (\dot{\theta}_{(N+1)} \Delta t + \ddot{\theta} \frac{\Delta t^2}{2})d \tag{20}$$

$$\dot{\psi}_{(N+1)} = \dot{\psi}_{(N)} + (\ddot{\psi} \Delta t)e \tag{21}$$

$$\psi_{(N+1)} = \psi_{(N)} + (\dot{\psi}_{(N+1)} \Delta t + \ddot{\psi} \frac{\Delta t^2}{2})e \tag{22}$$

$$\dot{\Phi}_{(N+1)} = \dot{\Phi}_{(N)} + (\ddot{\Phi} \Delta t)f \tag{23}$$

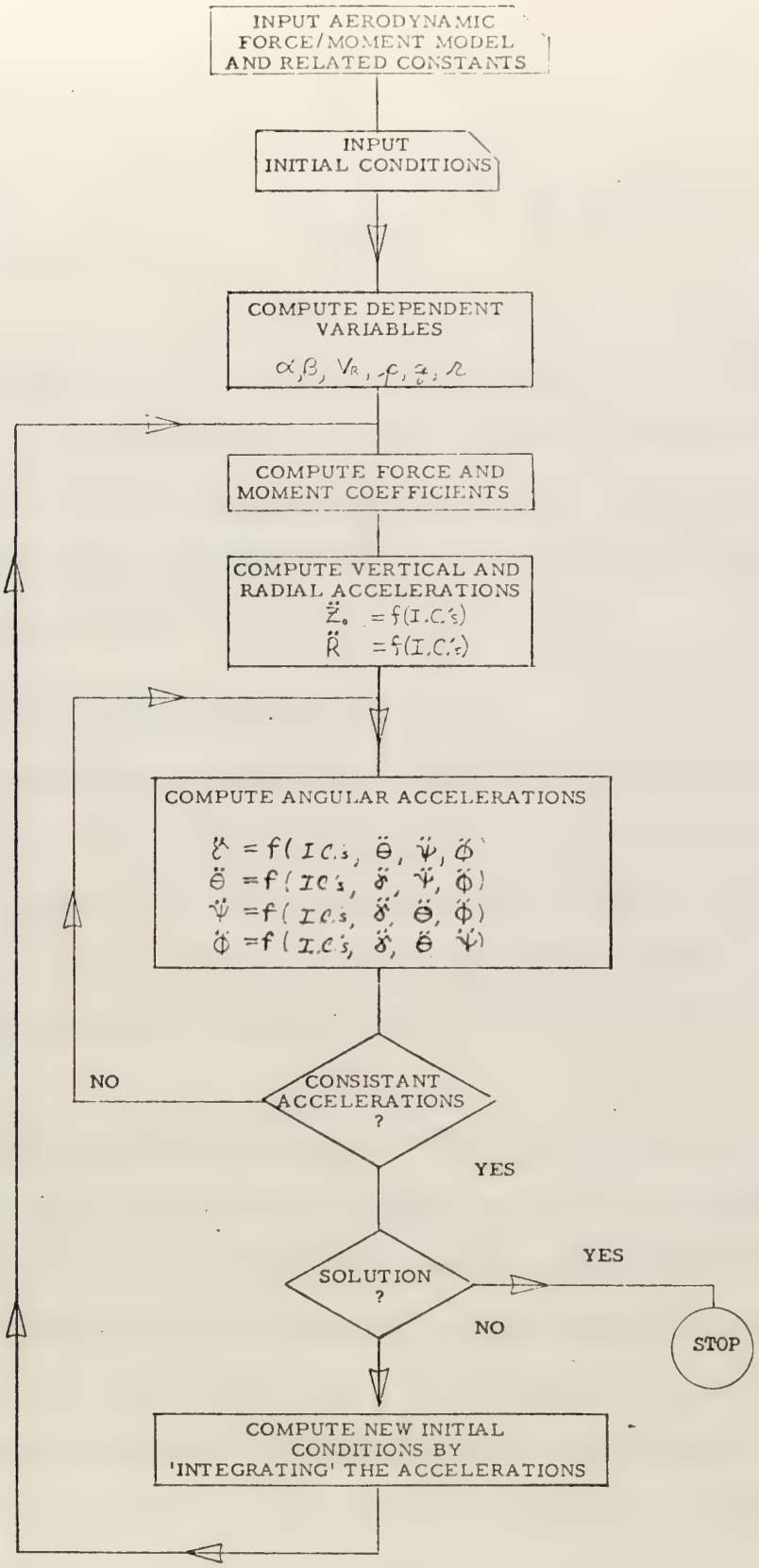



Figure 4

A Schematic Representation of the Integration Solution Method

$$\Phi_{(N+1)} = \Phi_{(N)} + \left( \dot{\Phi}_{(N+1)} \Delta t + \ddot{\Phi} \frac{\Delta t^2}{2} \right) f \qquad (24)$$

The constants (a, b, c, d, e, and f) are solution convergence factors which can be used to weigh the solution "rate" of a particular parameter. Also, one should note that the $(\dot{\theta})$, $(\dot{\psi})$, and $(\dot{\phi})$ expressions are used only in conjunction with the full equations. A schematic representation of the iterative process is depicted in Figure 4.

## H. THE GRADIENT SOLUTION METHOD

The existance of a dynamically stable equilibrium solution can be verified utilizing a set of equations which exclude all time dependent terms. A gradient solution method was developed to search for such solutions using the previously discussed short equations. The method utilized is commonly referred to as the optimal method of steepest descent.[4]

Conceptually, the solution space can be visualized as a series of peaks and valleys with solutions being the floor of any valley whose altitude is zero. The method is initiated at some 'geographic' position (initial conditions) where the "altitude" is measured (criterion function value). If the "altitude" were non-zero, the slope (gradient) of the "hill" would be measured and then the independent variables would be incrementally "marched" down the hill. The process continues until the independent variables reach the bottom of the "valley".

---

[4] Reference [7], page 275.

Mathematically, the criterion function is defined as the sum of the absolute values of the equation residuals. A solution is obtained when

$$S = \sum_{i=1}^{6} |S_i| = 0 \tag{25}$$

where $S_i$ are the equation residuals computed from the given independent variable vector, $X_i$ (i = 1, 6).

The rapid reduction of S to zero is accomplished by iteratively choosing an increment $\delta x_i$ such that $\delta S$ approaches -S where

$$\delta S = \sum_{i=1}^{6} \left(\frac{\partial S}{\partial X_i}\right) \delta x_i = -S \tag{26}$$

This is achieved by choosing $\delta x_i$ as the product of the criterion gradient and a gain constant K;

$$\delta x_i = K \frac{\partial S}{\partial X_i} \tag{27}$$

However, by noting that

$$\delta x_i = \sum_{i=1}^{6} \left(\frac{\partial S}{\partial X_i}\right)\left(K \frac{\partial S}{\partial X_i}\right) = -S \tag{28}$$

or

$$\delta S = K \sum_{i=1}^{6} \left(\frac{\partial S}{\partial X_i}\right)^2 = -S \tag{29}$$

K can be expressed as

$$K = \frac{-S}{\sum_{i=1}^{6} \left(\frac{\partial S}{\partial X_i}\right)^2} \tag{30}$$

The subsequent value of the $i^{th}$ element of the X vector becomes

$$X_{i_{(N+1)}} = X_{i_{(N)}} + \delta x_{i_{(N)}} \qquad i = 1, 2, \ldots, 6 \tag{31}$$

where the subscript n refers to a sequential iteration number.

$S_{n+1}$ is then computed as a function of $X_{n+1}$; if the condition

$$S_{N+1} < S_N \tag{32}$$

is satisfied, then the increment vector $\delta x_N$ is reapplied,

$$X_{N+2} = X_{N+1} + \delta x_N \tag{33}$$

and this 1-D search process is continued in the direction of the original gradient until

$$S_{N+1} > S_N \tag{34}$$

whereupon new values of the gradients are computed. Since the process avoids the computation of the partial derivatives whenever the condition expressed in equation (32) is satisfied, significant amounts of computer computational time is saved and hence, the name 'optimal' gradient solution method is achieved. Figure 5 includes a flow chart of this technique.

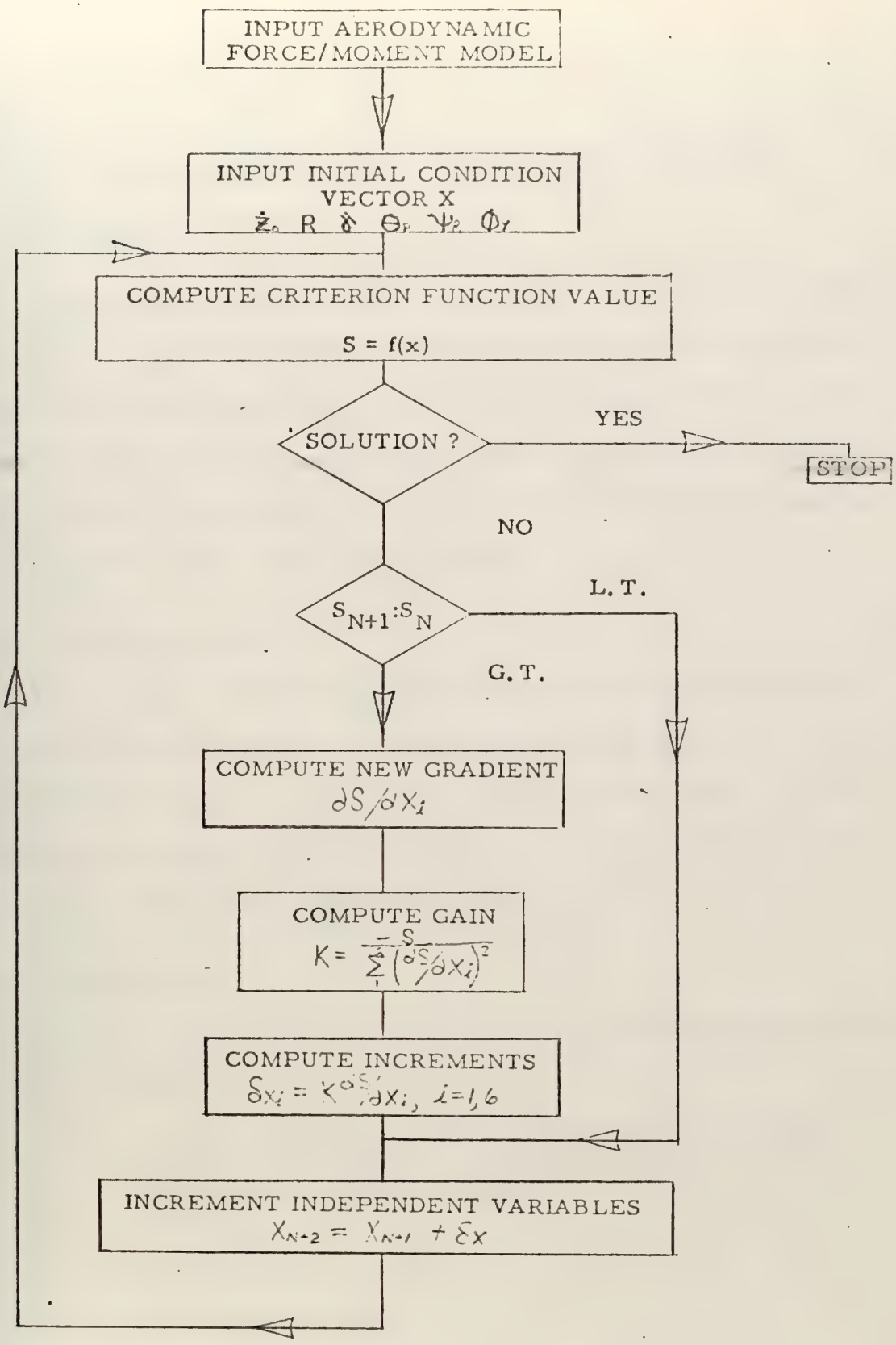



Figure 5

A Schematic Representation of the Gradient Solution Method

## III. AIRCRAFT SPIN SOLUTION COMPUTER PROGRAM

### A. GENERAL

This section presents briefly a general description of the digital computer program designed to solve the spin equations of motion for dynamically stable equilibrium solutions. A listing of the program as well as specific usage data and flow charts is presented in Appendix II. A program hierarchy is included as Figure 6 and an explanation of the program output is described by Figure 7.

### B. PROGRAM USAGE

The program reads the tabular force/moment coefficients and related aircraft constants and then commences to iteratively seek a solution based upon the first initial condition data set that is read. Upon termination of a particular solution, the program restarts by reading the next initial condition data set.

### C. INPUT DATA

The user must provide the following information to the program:

1. Type aircraft (F4, F111, etc.)
2. Tabular values of the force and moment coefficients/ stability derivatives (as described in Table II) and the corresponding values of angle of attack and sideslip angle
3. Aircraft mass
4. Atmospheric density
5. Mean aerodynamic chord
6. Gravitational constant
7. Total wing area

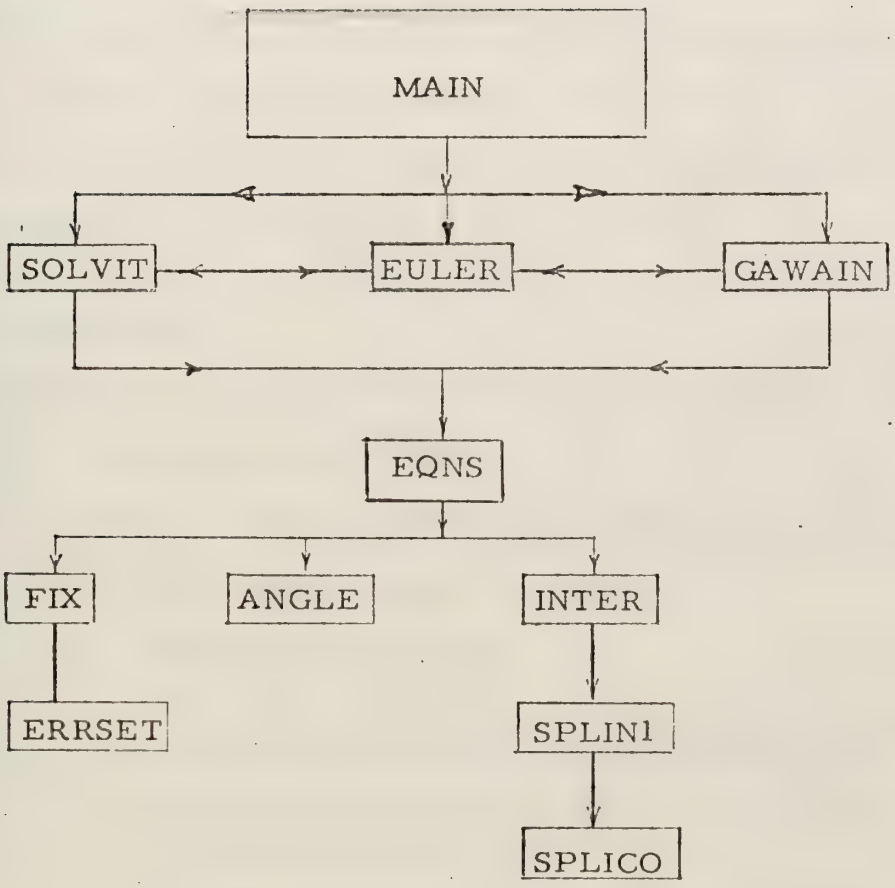



Figure 6

Airplane Spin Solution Program Hierarchy

8. Principal moments of inertia

9. Control deflections

10. Initial conditions of the independent variables

The user also must stipulate the solution method, equation form, solution criteria, time increment, output format and solution convergence factors.

The solution convergence factors available for the integration method, allow the user to accelerate or decelerate the solution "rate" of any of the six independent variables. For example, by equating a particular factor to zero, that variable will remain constant, thus reducing the number of degrees of system freedom.

## D. OUTPUT DATA

The printed output format is similar for the integration and gradient methods. The following is printed:

1. Listing of all input data under appropriate headings
2. Iterative results (integration method)
   a. iteration number (M1)
   b. the number of looping iterations required to obtain a consistent set of angular accelerations (M2)
   c. the independent variables with their associated first and second derivatives
   d. relative wind velocity, angle of attack, and sideslip angle
   e. sum of the absolute values of the accelerations
3. Iterative results (gradient method)
   a. iteration number (M1)
   b. the iteration number corresponding to the last gradient computation (M2)

INTEGRATION METHOD ITERATIVE OUTPUT

| M1 M2 | ALTITUDE (FT) | RATE (FT/SEC) | ACCEL | RADIUS (FEET) | ACCEL RATE | SPIN RATE (DEG/SEC) | ACCEL | THETA (DEG) | ACCEL RATE | PSI (DEG) | ACCEL RATE | PHI (DEG) | ACCEL RATE | ALPHA (DEG) | SUM VELOCITY | BETA (DEG) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -311.61 | | | 12.30 | | 77.39 | | -28.72 | | 8.62 | | 5.50 | | 61.35 | | -4.93 |
| 5 | | -.9080-02 | | | 0.7250 00 | | -.4040 00 | | -.1330 00 | | 0.4510 00 | | -.1570 00 | | 0.1880 01 | |
| | | | | | 0.7250-03 | | | | -.1330-03 | | 0.4510-03 | | -.1570-03 | | 0.3120 03 | |
| 2 | -311.61 | | | 12.30 | | 77.37 | | -28.72 | | 8.62 | | 5.50 | | 61.35 | | -4.93 |
| 2 | | -.9120-02 | | | 0.7120 00 | | -.4160 00 | | -.1320 00 | | 0.5250 00 | | -.2100 00 | | 0.2010 01 | |
| | | | | | 0.1440-02 | | | | -.2660-03 | | 0.9760-03 | | -.3670-03 | | 0.3120 03 | |

ITERATIONS REQUIRED TO ACHIEVE COMPATIBLE ANGULAR ACCELERATIONS

ITERATION NUMBER

SPIN RATE ( $\dot{\delta}$ )

ROLL ACCELERATION ( $\dot{\Psi}$ )

YAW RATE ( $\dot{\phi}$ )

ANGLE OF ATTACK

VELOCITY

SIDESLIP ANGLE

GRADIENT METHOD ITERATIVE OUTPUT

| M1 M2 | ALTITUDE (FT) | RATE (FT/SEC) | RESID DELX PRSX | RADIUS (FEET) | RESID DELX PRSX | SPIN RATE (DEG/SEC) | RESID DELX PRSX | THETA (DEG) | RESID DELX PRSX | PSI (DEG) | RESID DELX PRSX | PHI (DEG) | RESID DELX PRSX | ALPHA (DEG) | SUM VELOCITY K | BETA (DEG) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -311.61 | | | 12.30 | | 77.39 | | -28.72 | | 8.62 | | 5.50 | | 61.35 | | -4.93 |
| 1 | | -.9080-02 | | | 0.7250 00 | | 0.2210 00 | | -.8400 05 | | -.7120-01 | | -.8090 05 | | 0.1650 06 | |
| | | -.3120 01 | | | 0.1230 00 | | 0.1350-01 | | 0.5100-02 | | -.1370-01 | | 0.1550-01 | | 0.3120 03 | |
| | | 0.2590 04 | | | 0.1590 05 | | 0.8390 06 | | 0.5760 06 | | -.2980 05 | | 0.1370 07 | | -.5650-07 | |
| 2 | -311.61 | | | 12.30 | | 74.67 | | -26.87 | | 4.44 | | 5.22 | | 63.11 | | -0.65 |
| 1 | | 0.1330 01 | | | -.3670 00 | | 0.1410 00 | | 0.4920 05 | | -.1290-02 | | 0.1250 05 | | 0.6670 05 | |
| | | -.1460-03 | | | -.8970-03 | | -.4740-01 | | -.3260-01 | | 0.1680-02 | | -.7750-01 | | 0.3120 03 | |
| | | 0.2590 04 | | | 0.1590 05 | | 0.8390 06 | | 0.5760 06 | | -.2980 05 | | 0.1370 07 | | -.5650-07 | |
| 3 | -311.61 | | | 12.30 | | 71.96 | | -25.02 | | 0.25 | | 4.93 | | 65.00 | | 3.60 |
| 3 | | 0.2730 01 | | | -.1170 01 | | 0.8470-01 | | 0.1490 06 | | 0.1170 00 | | 0.1530 06 | | 0.3020 06 | |
| | | -.1460-03 | | | -.8970-03 | | -.4740-01 | | -.3260-01 | | 0.1680-02 | | -.7750-01 | | 0.3120 03 | |
| | | -.4890 04 | | | -.1290 05 | | -.6580 06 | | -.1310 07 | | 0.6600 05 | | 0.6530 05 | | -.1410-06 | |

ITERATION CORRESPONDING TO THE LAST GRADIENT COMPUTATION

ITERATION NUMBER

SPIN RATE ( $\dot{\delta}$ )

RESIDUAL OF THE $\dot{\delta}$ EQUATION

INCREMENT USED TO COMPUTE SUBSEQUENT VALUE OF $\dot{\delta}$

GRADIENT OF THE CRITERION FUNCTION WITH RESPECT TO $\dot{\delta}$; $\frac{\partial S}{\partial \dot{\delta}}$

CRITERION FUNCTION VALUE

GAIN FACTOR, K

Figure 7

Explanation of the Airplane Spin Solution Printed Output

c. the six independent variables with their corresponding residuals, incremental values, and partial derivatives

d. relative wind velocity, angle of attack, and sideslip angle

e. criterion value; the sum of the absolute values of the residuals

f. K; the solution gain factor

4. Summary output

Upon normal termination of the integration method, the iterative output is summarized. The listing includes the following data corresponding to every twenty-five iterations:

1. iteration number
2. independent variables
3. dependent variables ($\alpha$, $\beta$, $V_R$)

Also, the above data corresponding to the smallest sum of the absolute values of the accelerations is printed.

In the event of an abnormal termination, the program will print an appropriate message which informs the user as to the reason for the termination as well as the values of the offending variable(s).

## E. PROGRAM MODULE DESCRIPTION

The program consists of nine modules; one main and eight subroutines as described below:

1. MAIN

This routine is basically an input/output device. Tabulated data, fixed constants, and initial conditions are read into common memory, angular units are converted from degrees to radians; the optional output format is established and the first page is printed. The optional equation form and solution method, being determined,

a call is then made on the appropriate solution subroutine for subsequent computation.

2. SOLVIT

The integration solution method described in Section II is programmed into this subroutine. The iterative results are printed and appropriate summary data is presented on the last page of the printout.

3. GAWAIN

This subroutine incorporates the optimized gradient method as described in Section II. Its operation is similar to SOLVIT, however, no summary data is printed.

4. EQNS

Subroutine EQNS includes all of the independent and dependent equations. Calls are made primarily from either GAWAIN or SOLVIT. The current set of initial conditions is then used to compute the direction cosines, angle of attack and sideslip angle. Subroutine ANGLE is called to determine whether Alpha and Beta are within the appropriate definition limits. If so, the force and moment coefficients are then determined by interpolating the appropriate matrices using subroutines INTER, SPLIN1, and SPLICO.

The residuals are computed using either the short, modified or long form of the equations. A looping routine is utilized with the latter forms to obtain a consistent set of angular accelerations. Finally, upon computation of the sum of the absolute values of the residuals, execution is returned to the calling program.

5. EULER

The equations of motion require that angular units be in radians and that the reference Euler orientation angles be utilized. Subroutine EULER is used to convert computer variables to the output form of

angular units of degrees and ordered orientation angles. The subroutine is utilized by MAIN, GAWAIN and SOLVIT in order to allow output of the more usable ordered angles.

6. ANGLE

Subroutine ANGLE is utilized primarily by EQNS to check whether alpha or beta has exceeded the upper or lower limits of data definition. If the limits were exceeded, the iterative solution effort would be terminated and control returned to MAIN. ANGLE also reduces the absolute magnitude of angular displacements to less than $2\pi$ radians.

7. FIX

Subroutine FIX is used in conjunction with IBM supplied ERRSET to facilitate termination of an iterative solution in the event of an underflow, overflow or divide check while control is in EQNS. Control is returned to MAIN rather than effecting program termination as is the usual case.

8. INTER

Subroutine INTER is utilized by EQNS to interpolate the tabular force and moment coefficient data for specified values of alpha and beta. INTER calls SPLIN1 for the cubic interpolation of each matrix or vector.

9. SPLIN1 and SPLICO

Subroutines SPLIN1 and SPLICO were written by Mr. Rod Kure and are maintained by the NPS Computer Center Staff. Their primary function is to provide cubic interpolation of a given matrix or vector.

## V. AIRPLANE SPIN EQUATION VALIDATION

The airplane spin equations of motion were validated by establishing initial conditions for a power-off glide configuration and then applying them to the equations using the spin solution program.

A small computer program, TRIM, was prepared to enable the selection of a set of dynamically stable initial conditions. The TRIM program flowchart, the program listing and the sample output is included in Appendix D.

The condition of a power-off glide is mathematically described by three expressions:

$$TAN\,(\alpha-\theta) = C_D/C_L \tag{34}$$

$$W = \tfrac{1}{2}\rho V^2 S\left(C_L \cos(\alpha-\theta) + C_D \sin(\alpha-\theta)\right) \tag{35}$$

$$C_m = \frac{-C_{m_{\delta e}}}{\delta_e} \tag{36}$$

where w, $\alpha$, $\theta$, $\delta_e$, $C_l$, $C_d$, and $C_m$ are as depicted in Figure 8. The values for $C_l$ and $C_d$ are determined by transposing from the principal to the stability axis systems as indicated below:

$$C_L = -(C_z + C_{z_{\delta e}}\delta e)\cos\alpha + (C_x + C_{x_{\delta e}}\delta e)\sin\alpha \tag{37}$$

$$C_D = -(C_x + C_{x_{\delta e}}\delta e)\cos\alpha + (C_z + C_{z_{\delta e}}\delta e)\sin\alpha \tag{38}$$

Given constant values for $\alpha$, w, $\rho$, and $S$, one can solve equations (34)--(36) for the quantities $\delta_e$, $\theta$, and V. Subsequently the variables

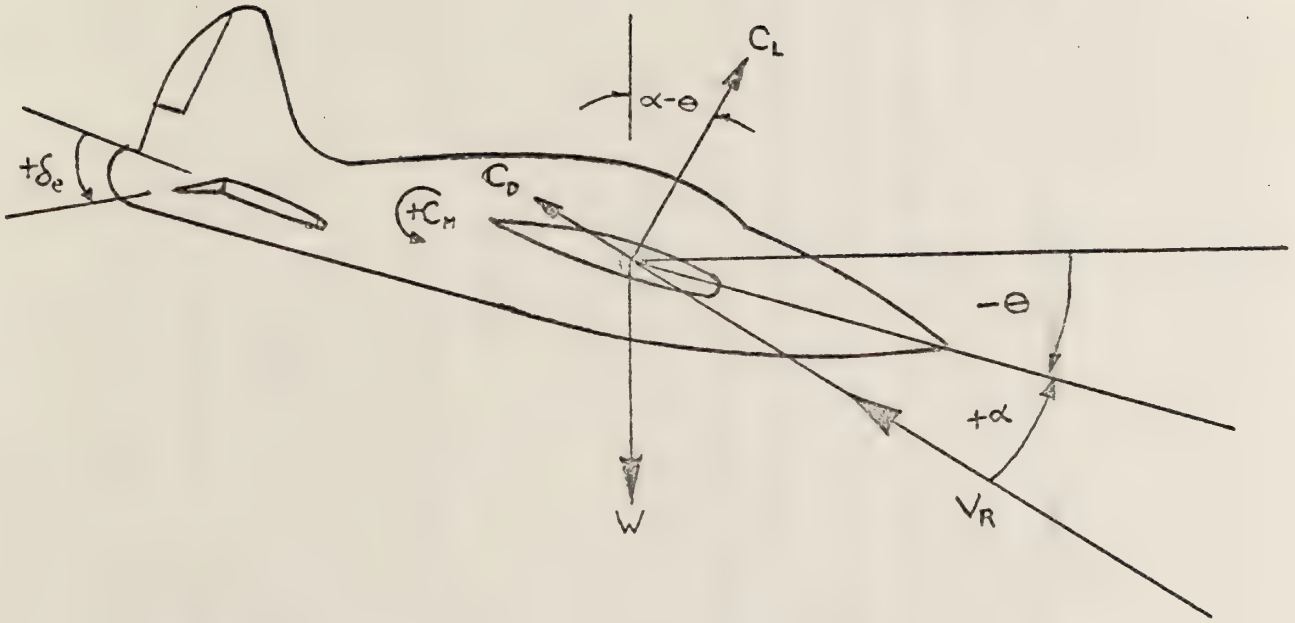



Figure 8

Schematic of the Aerodynamic Parameters for an Airplane in a Power-off Glide

$$\dot{Z}_o = V \sin(\alpha - \theta) \tag{39}$$

$$\dot{R} = V \cos(\alpha - \theta) \tag{40}$$

can then be determined.

The variables $Z_o$, R, and $\delta_e$ are thus established independent of the spin equations of motion but based upon the same aerodynamic force and moment model utilized by the spin solution program. Application of these initial conditions to the spin solution program containing Buehler's spin equations resulted in an immediate convergence whereby the sum of the absolute values of accelerations (equation residuals) was equal to 0.0182. Table III provides a listing of the pertinent data utilized to achieve the equation validation.

Table III

Pertinent Data Relating to Equation Validation

| | $Z_o$ | R | $\theta_p$ | $\alpha$ | $V_r$ | $\delta_e$ |
|---|---|---|---|---|---|---|
| TRIM Input: F-111 force and moment model; air density = $.1553 \times 10^{-2}$ | | | | | | |
| TRIM Output | 77.01[1] | -381.1[1] | .5763[1] | 12.0 | -388.8 | -5.534[1] |
| Integration Method Solution[2] | 76.99 | -381.0 | .57 | 12.0 | 389.0 | |

Notes: 1. Spin solution program input as per Table VIII
2. SUM = .0188

The spin solution program requires special care when exercising it for straight and level flight. Due to EULER restrictions about accepting quadrant II and III and cardinal orientation angles, the initial condition values should be applied as in Table IV below.

TABLE IV

Initial Conditions for Equation Validation

| VARIABLE | VALUE |
|---|---|
| $\rho, S, W$ | As used in TRIM Program |
| $\delta_e, \dot{Z}_o, \Theta, \dot{R}$ | As obtained from TRIM Output |
| $\delta_a, \delta_r, \dot{\gamma}$ | 0.0 |
| $\Psi, \Phi$ | .0001 |
| NSFLAG | 0 |

The results of the equation validation computer tests conclusively validate the spin equations of motion for the power-off glide configuration. This procedure can only be considered, at best, qualitative, for it does not test the equations in a time dependent environment and configuration. Complete validation would require the generation of time histories and subsequent comparison to base line data such as is included in Reference [2].

# V. NUMERICAL ANALYSIS AND CONCLUSIONS

## A. AERODYNAMIC FORCE AND MOMENT MODELS

The primary purpose of this thesis effort was to develop an aircraft spin solution computer program. During the program development, there were no idealized data available with known spin solutions, therefore the author utilized actual static wind tunnel data to provide an aerodynamic force and moment model. This was done in one case with the objective of ultimately comparing actual aircraft spin performance with computer generated predictions.

Two models were utilized. Tabular F-111 data was obtained from Reference [3] and F-4b data were extracted from curves presented in Reference [4]. In both cases appropriate values were multiplied by b/c in order to normalize to the common dimension utilized in the computer program, i.e., the mean aerodynamic chord. The above data are included in Appendix E.

## B. SOLUTION METHOD EVALUATION

### 1. Integration Solution Method

The integration solution method was the primary vehicle utilized to develop the spin equations of motion. As discussed in Section II, the primary advantage lies with the characteristic of the equation residuals being equivalent to the appropriate acceleration terms.

The method must be initiated with a set of initial conditions. Many sets of various initial conditions were utilized during the development but for the sake of brevity only the results from one arbitrary set will be discussed.

It was evident early in the development phase that the modified equations provided the greatest potential. The full equations were suitable only for verifying the stability of solutions previously obtained from the modified equations. When the full equations were used to evaluate a set of arbitrary initial conditions, the angular orientation rates rapidly increased to disproportionate values and induced tumbling motion which, in turn, resulted in a divergent solution. The modified equations, however, enable quasi-steady solutions and are responsive to small changes in the time increment and control deflections.

The solution paths generated by the integration method were characterized by an initially rapid convergence to a minimum value of the criterion function. The orientation values reorient to a nearly optimal configuration; the sink rate seeks the acceleration minimum; and the spin rate and radius effect trade-offs to achieve a consistent velocity value. Occasionally, the solution path became divergent in that the program was unable to find a consistent set of coupled accelerations. This was evidenced by the printout of the number of iterations required by the acceleration "do loop" within Subroutine EQNS. The "do loop" limit is set at twenty iterations to minimize computer time, however, only two iterations are usually needed for a stable solution path. Divergence is characterized by large and erratic changes in the independent variables such that alpha or beta exceed their respective fields of definition, thus resulting in solution termination. Solution path divergence can be minimized through use of a small time increment. The author determined that a range between .001 and .05 seconds proved optimal. The solution convergence factors were not utilized sufficiently to merit discussion.

Once the initial convergence was achieved, the individual solution paths seemed to vary in a seemingly arbitrary manner. While one variable tended toward a minimum acceleration value, other variables were producing increasing values. The variables continued to "wander"; changing magnitude and sign until either computer time was exhausted or the alpha or beta limits were exceeded.

All solution efforts using F-4 data with pro-spin control deflections resulted in solution paths which exceeded the sideslip angle limits. Inclusion of rotary balance data to compensate for differential sideslip angles along the fuselage might inhibit such yawing action.

Utilization of the integration method requires that the user learn to iterate the output to optimize the achievement of a final solution. This process is as described below:

a. An arbitrary set of initial conditions is applied to the program to initiate a 500 iteration solution effort. This set of initial conditions should be such so as to approximate the anticipated spin solution mode.

b. The initial condition set which yields the best convergence is modified by substituting the value for $\dot{Z}_o$ corresponding to the minimum $\ddot{Z}_o$ in the 500 iteration output. An alternative is to substitute the set of three orientation angles corresponding to the minimum absolute sum of the orientation angle accelerations. Similarly, R and $\dot{\chi}$ can be substituted. The resultant set of revised initial conditions is then used to initiate subsequent solution efforts of shorter (150 iterations) duration.

c. The process is continued until a best solution is obtained.

d. The final solution is tested for stability by using the full equations with a small time increment.

Figure 9 and Table V illustrate a typical solution effort using an arbitrary set of initial conditions.

The integration method results lead to a number of conclusions:

a. The F-111 aerodynamic force and moment model precluded finding stable equilibrium solutions. This can be attributed to any of four reasons: (1) there were no truly steady state solutions; only oscillatory modes exist; (2) the "right" set of initial conditions was never employed and therefore a steady state solution path was not intercepted; (3) the steady state modes were lightly damped and therefore, may be very slow in convergence; the program wasn't exercised long enough to find the solution; and (4) the integration solution method itself precludes uniform convergence of all degrees of freedom. This would be due to self-excitation of the non-linear coupled acceleration terms.

b. The F-4 data do not model the actual aircraft since all initial conditions yielded solution paths with divergent yaw rates. The actual aircraft does not exhibit this trait and, in fact, demonstrates several steady and quasi-steady spin modes.

c. The method is cumbersome to use because it requires that the user iteratively adjust inputed initial conditions in order to achieve a solution.

d. The radius rate terms incorporated in the modified equations enables the determination of quasi-steady state solutions using the F-111 data.

e. The orientation rate terms in the full equations were not compatible with the course integration techniques of this method.

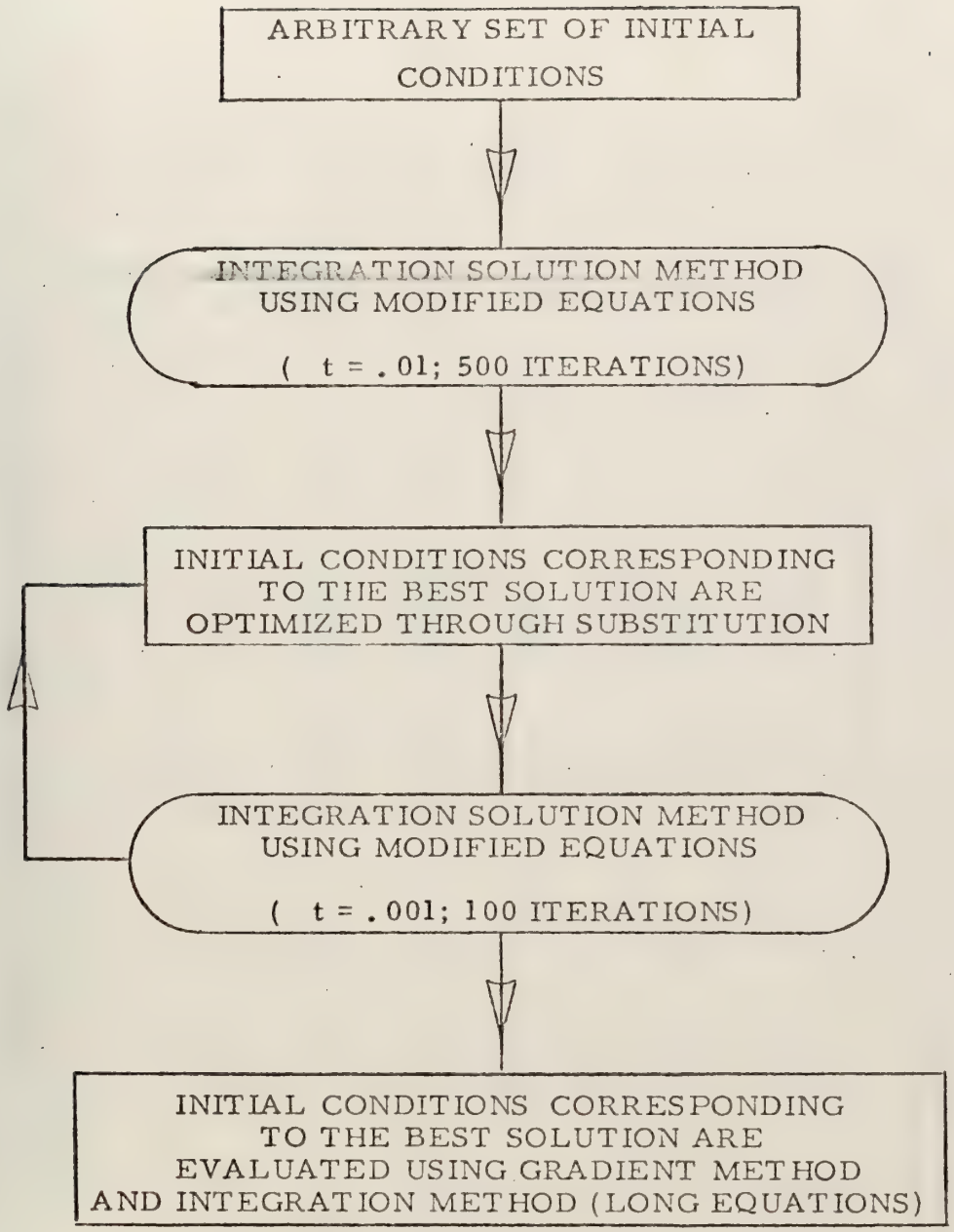

ARBITRARY SET OF INITIAL CONDITIONS
INTEGRATION SOLUTION METHOD USING MODIFIED EQUATIONS
( t = .01; 500 ITERATIONS)
INITIAL CONDITIONS CORRESPONDING TO THE BEST SOLUTION ARE OPTIMIZED THROUGH SUBSTITUTION
INTEGRATION SOLUTION METHOD USING MODIFIED EQUATIONS
( t = .001; 100 ITERATIONS)
INITIAL CONDITIONS CORRESPONDING TO THE BEST SOLUTION ARE EVALUATED USING GRADIENT METHOD AND INTEGRATION METHOD (LONG EQUATIONS)

Figure 9

Schematic of a Typical Integration Solution Effort

TABLE V

Results of a Typical Integration Solution Effort

| RUN # | $\dot{Z}_o$ | R | $\dot{R}$ | $\dot{\gamma}$ | $\Theta_p$ | $\Psi_R$ | $\Phi_v$ | SUM |
|---|---|---|---|---|---|---|---|---|
| 1 | -400.0 | 50.0 | 0.0 | 30.0 | -30.0 | 10.0 | -5.0 | -- |
| | -372.36 | 21.37 | 31.8 | 67.68 | -29.38 | 9.56 | -1.22 | 25.41 |
| 2 | -332.36 | 21.37 | 0.0 | 67.68 | -29.38 | 9.56 | -1.22 | -- |
| | -329.98 | 12.30 | -11.7 | 79.13 | -28.90 | 8.93 | 3.74 | 7.89 |
| 3 | -317.33 | 12.30 | 0.0 | 79.13 | -28.90 | 8.93 | 3.74 | -- |
| | -317.26 | 12.30 | .063 | 77.87 | -28.88 | 8.94 | 3.78 | 4.61 |
| 4 | -317.26 | 12.30 | 0.0 | 77.87 | -28.72 | 8.62 | 5.50 | -- |
| | -317.25 | 12.30 | .0027 | 77.63 | -28.72 | 8.62 | 5.50 | 2.53 |
| 5 | -314.61 | 12.30 | 0.0 | 77.39 | -28.72 | 8.62 | 5.50 | -- |
| | -314.61 | 12.30 | .003 | 77.39 | -28.72 | 8.62 | 5.50 | 2.06 |
| 6 | -311.61 | 12.30 | 0.0 | 77.39 | -28.72 | 8.62 | 5.50 | -- |
| | -311.61 | 12.30 | .0003 | 77.39 | -28.72 | 8.62 | 5.50 | 1.88 |
| 7[1] | -311.61 | 12.30 | n/a | 77.39 | -28.72 | 8.62 | 5.50 | -- |
| | -311.61 | 12.30 | n/a | 68.99 | -35.30 | 14.58 | 6.15 | $29.2 \times 10^4$ |

Upper values correspond to the inputted initial conditions while the lower values correspond to the best solution (minimum value of the criterion function: SUM).

NOTES:
1. Run #7 utilized the gradient solution method.
2. The inputted initial conditions correspoinding to Run #6 were also applied to the long equations using the integration method but the solution path was diverging when computation was terminated.
3. F-111 aerodynamic model data was utilized with full pro-spin control deflections.

f. The computer program needs to be modified to incorporate a subroutine which will compute the spin radius as well as the cylindrical inertial radius as in the present version.

2. Gradient Solution Method

The gradient solution method and its derivatives from optimal control theory have, perhaps, the greatest potential in the area of steady-state spin prediction. The method programmed by the author can only be considered crude when compared to some of the more sophisticated methods which utilize hybrid applications.

The results obtained from a typical F-111 gradient solution effort provide the basis for evaluation of this method as well as yield insight into the F-111 solution space. Table VI includes a summary of these results. One should note that there was an initial convergence of the angular variables in the first fifty iterations. After iteration #75, the solution stabilized but the criterion value alternated between 27.0 and 102.0; this was a result of the alternating residuals of the pitch and yaw equations. The sensitivity of the solution is extreme when one considers that a 0.4% change in the absolute sum of the independent variables results in a 2,740% reduction in the absolute sum of the acceleration. Specifically, the alternating changes in the $\Theta$ and $\Phi$ residuals are the result of incremental changes in the $\Theta$ and $\Phi$ values of only $10^{-4}$ radians!

The solution convergence becomes static due to the characteristically large gradients ($10^{3}$-$10^{7}$) and resultant small incremental step sizes. All gradient solutions attempted, resulted in no significant change in the values for $\dot{Z}_o$ and R parameters.

Two conclusions can be made from the gradient solution method results:

TABLE VI

Summary of a F-111 Gradient Solution Effort

| | $\dot{Z}_o$ | R | $\dot{\gamma}$ | $\Theta_p$ | $\Psi_R$ | $\Phi_Y$ | SUM |
|---|---|---|---|---|---|---|---|
| Iteration Number | | | Iterative Results | | | | |
| 0 | -200.0 | 100.0 | 10.0 | -31.0 | 5.0 | -5.0 | $7.38 \times 10^3$ |
| 25 | -200.0 | 100.0 | 9.42 | -31.5 | 5.35 | -5.08 | $2.63 \times 10^3$ |
| 50 | -200.0 | 100.0 | 9.39 | -31.38 | 5.34 | -5.05 | $0.73 \times 10^2$ |
| 75 | -200.0 | 100.0 | 9.39 | -31.37 | 5.34 | -5.05 | $1.02 \times 10^2$ |
| 76 | -200.0 | 100.0 | 9.39 | -31.37 | 5.34 | -5.05 | $.279 \times 10^2$ |
| 100 | -200.0 | 100.0 | 9.39 | -31.37 | 5.34 | -5.05 | $1.02 \times 10^2$ |
| 101 | -200.0 | 100.0 | 9.39 | -31.37 | 5.34 | -5.05 | $.270 \times 10^2$ |
| | Other Representative Values as per the Indicated Equations[1] | | | | | | |
| | $\dot{Z}_o$ | R | $\dot{\gamma}$ | $\Theta$ | $\Psi$ | $\Phi$ | |
| Residual ($S_i$) | $-1.9 \times 10^4$ | -6.79 | 0.24 | $.34^2$ | .186 | $-.39^3$ | |
| Increment ($\delta_{x_i}$) | $3.0 \times 10^{-8}$ | $0.5 \times 10^{-7}$ | $3.7 \times 10^{-5}$ | $\pm 5.0 \times 10^{-4}$ | $-1.0 \times 10^{-6}$ | $\pm 3.5 \times 10^{-5}$ | |
| Gradient ($\frac{\partial S}{\partial X_i}$) | $6.1 \times 10^2$ | $-1.0 \times 10^3$ | $-.74 \times 10^6$ | $1.0 \times 10^8$ | $2.0 \times 10^6$ | $.71 \times 10^6$ | |

NOTES: 1. Data is representative from iteration #25 to termination at #350.
2. The $\Theta$ residual alternated between 0.24 and $\pm$ 61.2.
3. The $\Phi$ residual alternated between 0.39 and $\pm$ 14.8.

a. The method was unable to find a steady state solution; however, in one case, it was able to effect a reduction of the criterion function value by four orders of magnitude.

b. The F-111 solution path is highly sensitive to small changes in the independent variable values.

## C. ANALYSIS OF THE F-111 AND F-4 STEADY-STATE SOLUTION SPACES

The integration and gradient solution methods were both used to analyze the F-111 and F-4 steady state solution spaces. Initial conditions were generated by a 'Criterion Function Search Program'. This program evaluated criterion function values from a course grid of 186,624 sets of initial conditions for each aircraft.

One can visualize the solution space as a geographical relief with the 'solutions' occurring at the valley floors of zero elevation. The search program essentially measured the 'altitudes' (criterion function values) in this area and found only a very few points whose elevations were less than $10^4$. Specifically, in the case of the F-4, only eleven sets of initial conditions generated criterion function values less than $10^4$ and only one set of F-111 initial conditions met this criteria. The numerical results of the criterion function search program are included in Table VII.

These generated initial conditions were evaluated using the gradient and integration solution methods. The results are extremely significant because they reveal that it is highly unlikely that a dynamically stable equilibrium solution exists. These solution results are summarized in Figure 10 and Table VIII.

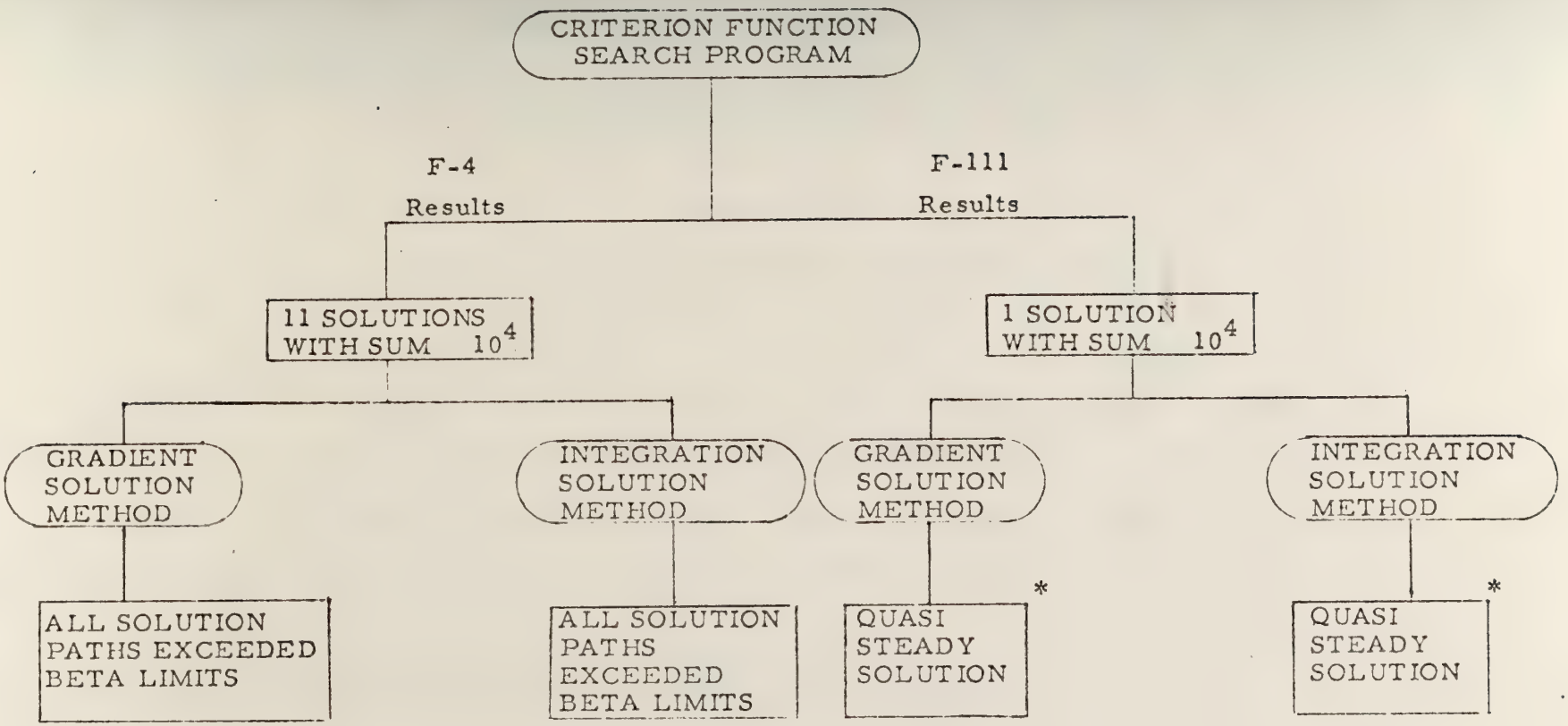
The two conclusions which can be made are:

TABLE VII

Initial Conditions Generated by the Criterion Search Program Which Yielded Criterion Values Less Than $10^4$

| A/C | $\dot{Z}_o$ | R | $\dot{\delta}$ | $\Theta_p$ | $\Psi_R$ | $\Phi_Y$ | $\alpha$ | $\beta$ | SUM* |
|---|---|---|---|---|---|---|---|---|---|
| F-111 | -200.0 | 100.0 | 10.0 | -31.0 | 5.0 | -5.0 | 59.48 | -22.42 | .738 |
| F-4 | -150.0 | 25.0 | 10.0 | -41.0 | 5.0 | -35.0 | 44.42 | -39.84 | .377 |
| F-4 | -150.- | 25.0 | 10.0 | -51.0 | 25.0 | -15.0 | 43.91 | -38.81 | .816 |
| F-4 | -150.0 | 25.0 | 20.0 | -31.0 | 25.0 | - 5.0 | 64.18 | -39.44 | .842 |
| F-4 | -150.- | 25.0 | 30.0 | -21.0 | 5.0 | - 5.0 | 69.64 | -30.16 | .640 |
| F-4 | -150.0 | 100.0 | 10.0 | -31.0 | 15.0 | - 5.0 | 62.03 | -35.31 | .834 |
| F-4 | -150.0 | 175.0 | 10.0 | -31.0 | 5.0 | - 5.0 | 59.69 | -33.32 | .958 |
| F-4 | -150.0 | 175.0 | 10.0 | -41.0 | 15.0 | - 5.0 | 52.46 | -37.27 | .761 |
| F-4 | -200.0 | 25.0 | 10.0 | -41.0 | 5.0 | -35.0 | 44.60 | -39.49 | .705 |
| F-4 | -200.0 | 175.0 | 10.0 | -41.0 | 5.0 | -25.0 | 45.27 | -39.88 | .559 |
| F-4 | -300.0 | 250.0 | 10.0 | -41.0 | 5.0 | -25.0 | 45.38 | -39.49 | .513 |
| F-4 | -350.0 | 175.0 | 30.0 | -31.0 | 5.0 | - 5.0 | 59.79 | -38.08 | .332 |

*All values x $10^4$




*Refer to Table IV for results.

Figure 10

Criterion Function Search Program Results

TABLE VIII

Gradient and Integration Results from Criterion Search
Generated Initial Conditions (F-111)

| METHOD | $\dot{Z}_o$ | R | $\dot{\gamma}$ | $\theta_P$ | $\psi_R$ | $\phi_Y$ | SUM |
|---|---|---|---|---|---|---|---|
| Gradient | -200.0 | 100.0 | 9.39 | -31.37 | 5.34 | 5.05 | 27.02 |
| Integration | -258.9 | 35.46 | 37.55 | -30.84 | 4.09 | -1.51 | 14.38 |

The above solutions correspond to the minimum value of SUM.

1. The static wind tunnel data utilized to form the basis for the F-111 and F-4 solution spaces are of such a nature as to preclude finding a dynamically stable equilibrium solution. There was no solution convergence to a criterion value of less than 25.0.

2. The aerodynamic data utilized do not adequately describe the actual aerodynamic force and moment field experienced by the actual aircraft. This is evidenced by the failure to find even a quasi-steady solution similar to flight test results.

The above implications are not surprising considering that the aerodynamics of the spin problem involve low speed, three-dimensional stalled flow and, as such, forces and moments are nonlinear. Thus the superposition of the separate effects of triaxial rotation, control deflection, and other static wind tunnel data cannot be completely justified analytically. The only wind tunnel test technique that presently appears to offer any possibility of yielding satisfactory aerodynamic data for the spin is the rotary balance force and moment measuring apparatus which is described in Reference [6].

## VI. SUMMARY OF THE MAJOR CONCLUSIONS

The criterion function search revealed that, for the F-111 and F-4 aerodynamic force and moment models utilized in this study, it was highly improbable that a dynamically stable equilibrium solution existed.

The airplane spin equations described in Reference [1] were qualitatively validated. Quantitative verification is needed and it is recommended that the equations be cast in a form compatible with a time history analysis in order that the resultant data may be compared with the base-line spin data included in Reference [2].

The integration solution method is too cumbersome to be effectively utilized although it can provide information relating to quasi-steady solutions once the user has learned to iteratively manipulate the initial conditions.

The gradient solution method demonstrated its potential by effecting significant reductions in criterion function values to quasi-steady state solutions.

## APPENDIX A

### FULL SPIN EQUATIONS

The following equations constitute the full equations of motion for a spinning airplane as written in a cylindrical coordinate system.

Z Equation

$$m\ddot{Z}_o + mg = -\left(C_{F_x}\alpha_{13} + C_{F_y}\alpha_{23} + C_{F_z}\alpha_{33}\right)\cdot$$

$$\frac{\rho S}{2}\left\{(\dot{\gamma}R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2\right\}$$

R Equation

$$m\ddot{R} - m\dot{\gamma}^2 R = -\left(C_{F_x}\alpha_{11} + C_{F_y}\alpha_{21} + C_{F_z}\alpha_{31}\right)\cdot$$

$$\frac{\rho S}{2}\left\{(\dot{\gamma}R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2\right\}$$

$\ddot{\gamma}$ Equation

$$\ddot{\gamma} = \left\{ mR^2 + I_x \alpha_{13}^2 + I_y \alpha_{23}^2 + I_z \alpha_{33}^2 \right\}^{-1} \cdot$$

$$\Big[ -2 \left\{ m \dot{\gamma} R \dot{R} + (\dot{\psi} + \dot{\gamma}) \left[ (I_x - I_y) \cos\phi \, \alpha_{13} \dot{\phi} \right.\right.$$

$$\left.\left. + (I_x \sin^2\phi + I_y \cos^2\phi - I_z) \alpha_{33} \dot{\theta} \right] \sin\theta \right\}$$

$$- \ddot{\psi} (I_x \alpha_{13}^2 + I_y \alpha_{23}^2 + I_z \alpha_{33}^2)$$

$$- \ddot{\theta} (I_x - I_y) \sin\phi \cos\phi \cos\theta \, \dot{\theta}^2$$

$$+ \left\{ \bar{c} (C_{m_x} \alpha_{13} + C_{m_y} \alpha_{23} + C_{m_z} \alpha_{33}) \right.$$

$$\left. - R (C_{F_x} \alpha_{12} + C_{F_y} \alpha_{22} + C_{F_z} \alpha_{32}) \right\} \cdot$$

$$\left. \frac{\rho S}{2} \left\{ (\dot{\gamma} R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2 \right\} \right]$$

$\ddot{\theta}$ Equation

$$\ddot{\theta} = \left\{ I_x \cos^2\phi + I_y \sin^2\phi \right\}^{-1} \Big[ (\dot{\psi} + \dot{\gamma}) \left\{ I_x \sin^2\phi \right.$$

$$\left. + I_y \cos^2\phi - I_z \right\} \left\{ \dot{\phi} + (\dot{\psi} + \dot{\gamma}) \cos\theta \right\} \sin\theta$$

$$+ 2 (I_x - I_y) (\ddot{\gamma} + \ddot{\psi}) \sin\theta \sin\phi \cos\phi$$

$$+ \left\{ C_{m_x} \cos\phi - C_{m_y} \sin\phi \right\} \frac{\rho S \bar{c}}{2} \left\{ (\dot{\gamma} R)^2 \right.$$

$$\left. + (\dot{R})^2 + (\dot{Z}_o)^2 \right\} \Big]$$

$\Psi$ Equation

$$\ddot{\psi} = \left\{ I_x \alpha_{13}^2 + I_y \alpha_{23}^2 + I_z \alpha_{33}^2 \right\}^{-1} \Big[ -2 (\dot{\psi} + \dot{\gamma}) \cdot$$

$$\left\{ (I_x - I_y) \alpha_{13} \alpha_{23} \dot{\phi} + (I_x \sin^2\phi + I_y \cos^2\phi \right.$$

$$\left. - I_z ) \sin\theta \cos\theta \dot{\theta} \right\} - (I_x - I_y) \dot{\theta}^2 \alpha_{13} \cos\phi$$

$$- I_z \ddot{\phi} \alpha_{33} - \dot{\gamma} (I_x \alpha_{13}^2 + I_y \alpha_{23}^2 + I_z \alpha_{33}^2)$$

$$+ \left\{ (I_x - I_y)(\sin^2\phi - \cos^2\phi) + I_z \right\} \sin\theta \dot{\phi} \dot{\theta}$$

$$- (I_x - I_y) \sin\phi \cos\phi \cos\theta \dot{\theta}^2$$

$$+ \left\{ C_{m_x} \alpha_{13} + C_{m_y} \alpha_{23} + C_{m_z} \alpha_{33} \right\} \cdot$$

$$\left. \frac{\rho S \bar{c}}{2} \left\{ (\dot{\gamma} R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2 \right\} \right]$$

$\Phi$ Equation

$$\ddot{\phi} = I_z^{-1} \Big[ -I_z (\ddot{\psi} + \ddot{\gamma}) \alpha_{33} + (I_x - I_y) \left\{ (\dot{\psi} + \dot{\gamma}) \alpha_{13} \right.$$

$$\left. + \dot{\theta} \cos\phi \right\} \left\{ (\dot{\psi} + \dot{\gamma}) \alpha_{23} - \dot{\theta} \sin\phi \right\}$$

$$+ (C_{m_z}) \frac{\rho S \bar{c}}{2} \left\{ (\dot{\gamma} R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2 \right\} \Big]$$

$Z_o$ Equation:

$$\ddot{Z}_o = -g - (C_{Fx}\alpha_{13} + C_{Fy}\alpha_{23} + C_{Fz}\alpha_{33}) \cdot \frac{\rho S}{2M}\left\{(\dot{\gamma}R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2\right\}$$

R Equation:

$$\ddot{R} = \dot{\gamma}^2 R - (C_{Fx}\alpha_{11} + C_{Fy}\alpha_{21} + C_{Fz}\alpha_{31}) \cdot \frac{\rho S}{2M}\left\{(\dot{\gamma}R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2\right\}$$

$\gamma$ Equation:

$$\ddot{\gamma} = \left\{MR^2 + I_x\alpha_{13}^2 + I_y\alpha_{23}^2 + I_z\alpha_{33}^2\right\}^{-1}\left[-I_x\alpha_{13}^2\right.$$

$$-(I_x - I_y)\dot{\theta}\dot{\phi}\sin\theta\sin\phi\cos\phi - I_z\dot{\phi}\alpha_{33}$$

$$-2MR\dot{R}\dot{\gamma} + \left\{\bar{c}\,(C_{mx}\alpha_{13} + C_{my}\alpha_{23} + C_{mz}\alpha_{33})\right.$$

$$\left.\left.-R(C_{Fx}\alpha_{13} + C_{Fy}\alpha_{12} + C_{Fz}\alpha_{32})\right\}\right] \cdot \frac{\rho S}{2}\left\{(\dot{\gamma}R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2\right\}$$

$\Theta$ Equation:

$$\ddot{\Theta} = \left\{ I_x \cos^2\Phi + I_y \sin^2\Phi \right\}^{-1} \left[ -(I_x - I_y)(\dot{\Psi} + \dot{\gamma}_1)\sin\Phi \sin\Theta \sin\Phi \right.$$

$$+ (I_x \sin^2\Phi + I_y \cos^2\Phi - I_z)\dot{\gamma}_1^2 \sin\Theta \cos\Theta + \left\{ C_{m_x} \cos\Phi \right.$$

$$\left. \left. - C_{m_y} \sin\Phi \right\} \frac{\rho S \bar{c}}{2} \left\{ (\dot{\gamma}_1 R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2 \right\} \right]$$

$\Psi$ Equation:

$$\ddot{\Psi} = \left\{ I_x \alpha_{13}^2 + I_y \alpha_{23}^2 + I_z \alpha_{33}^2 \right\}^{-1} \left[ -(I_x \alpha_{13}^2 + I_y \alpha_{23}^2 \right.$$

$$+ I_z \alpha_{33}^2)\ddot{\gamma}_1 - (I_x - I_y)\dot{\Theta}\dot{\Phi} \sin\Theta \sin\Phi \cos\Phi$$

$$- I_z \dot{\Phi}\dot{\Theta} \cos\Theta + \left\{ C_{m_x}\alpha_{13} + C_{m_y}\alpha_{23} + C_{m_z}\alpha_{33} \right\} \cdot$$

$$\left. \frac{\rho S \bar{c}}{2} \left\{ (\dot{\gamma}_1 R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2 \right\} \right]$$

$\Phi$ Equation:

$$\ddot{\Phi} = \left( \frac{I_x - I_y}{I_z} \right) \dot{\gamma}_1^2 \sin^2\Theta \sin\Phi \cos\Phi - (\ddot{\gamma}_1 + \ddot{\Psi})\cos\Theta$$

$$+ C_{m_z} \frac{\rho S \bar{c}}{2 I_z} \left\{ (\dot{\gamma}_1 R)^2 + (\dot{R})^2 + (\dot{Z}_o)^2 \right\}.$$

SHORT EQUATIONS

$Z_o$ Equation:

$$2gm + \rho S\left\{(\dot{\gamma}R)^2 + (\dot{Z}_o)^2\right\}\left\{C_{F_X}\alpha_{13} + C_{F_Y}\alpha_{23} + C_{F_Z}\alpha_{33}\right\} = 0$$

R Equation:

$$2\dot{\gamma}^2 Rm - \rho S\left\{(\dot{\gamma}R)^2 + (\dot{Z}_o)^2\right\}\left\{C_{F_X}\alpha_{11} + C_{F_Y}\alpha_{21} + C_{F_Z}\alpha_{31}\right\} = 0$$

$\gamma$ Equation:

$$C_{F_X}\alpha_{12} + C_{F_Y}\alpha_{22} + C_{F_Z}\alpha_{32} = 0$$

$\Theta$ Equation:

$$2\left(I_X \sin^2\phi + I_Y \cos^2\phi - I_Z\right)\dot{\gamma}^2 \sin\Theta\cos\Theta$$
$$+ \rho S\bar{c}\left\{(\dot{\gamma}R)^2 + \dot{Z}_o^2\right\}\left\{C_{m_X}\cos\phi - C_{m_Y}\sin\phi\right\} = 0$$

$\psi$ Equation:

$$C_{m_X}\alpha_{13} + C_{m_Y}\alpha_{23} + C_{m_Z}\alpha_{33} = 0$$

$\phi$ Equation:

$$2\left(I_X - I_Y\right)\dot{\gamma}\sin^2\theta\sin\phi\cos\phi$$
$$+ \rho S\bar{c}\left\{(\dot{\gamma}R)^2 + \dot{Z}_o^2\right\}C_{m_Z} = 0$$

## APPENDIX B

## AIRCRAFT SPIN SOLUTION PROGRAM

### A. GENERAL

This appendix is intended for the Fortran programmer who is a potential user of the program. The included source listing contains an extensive introductory comment section with which the reader should familiarize himself prior to reading the other sections of this appendix.

### B. DATA INPUT

The composition of the data deck is described in the source listing and also depicted in Figure B-1. One should particularly note the composition of data sets #2--#43. The first card of each of the aforementioned sets "tells" the program which tabular data is to follow, its literal name, and whether the data set is a matrix, a vector, or a null set. The subsequent cards of each data set contain the tabular data except where no data is available (null set); in which case, the data set consists only of the first card. Each force/moment coefficient matrix is characterized by a unique integer assignment (NA) as specified in Table B-I. NA is the first value of each of the 42 coefficient data sets. There must be exactly 42 "NA" cards however there is no restriction as to their sequence.

Each solution effort is initiated based upon the initial conditions and associated program variables provided by data set #45. An option is provided (integration method only) for the program variables to remain constant for subsequent solution efforts; only the initial

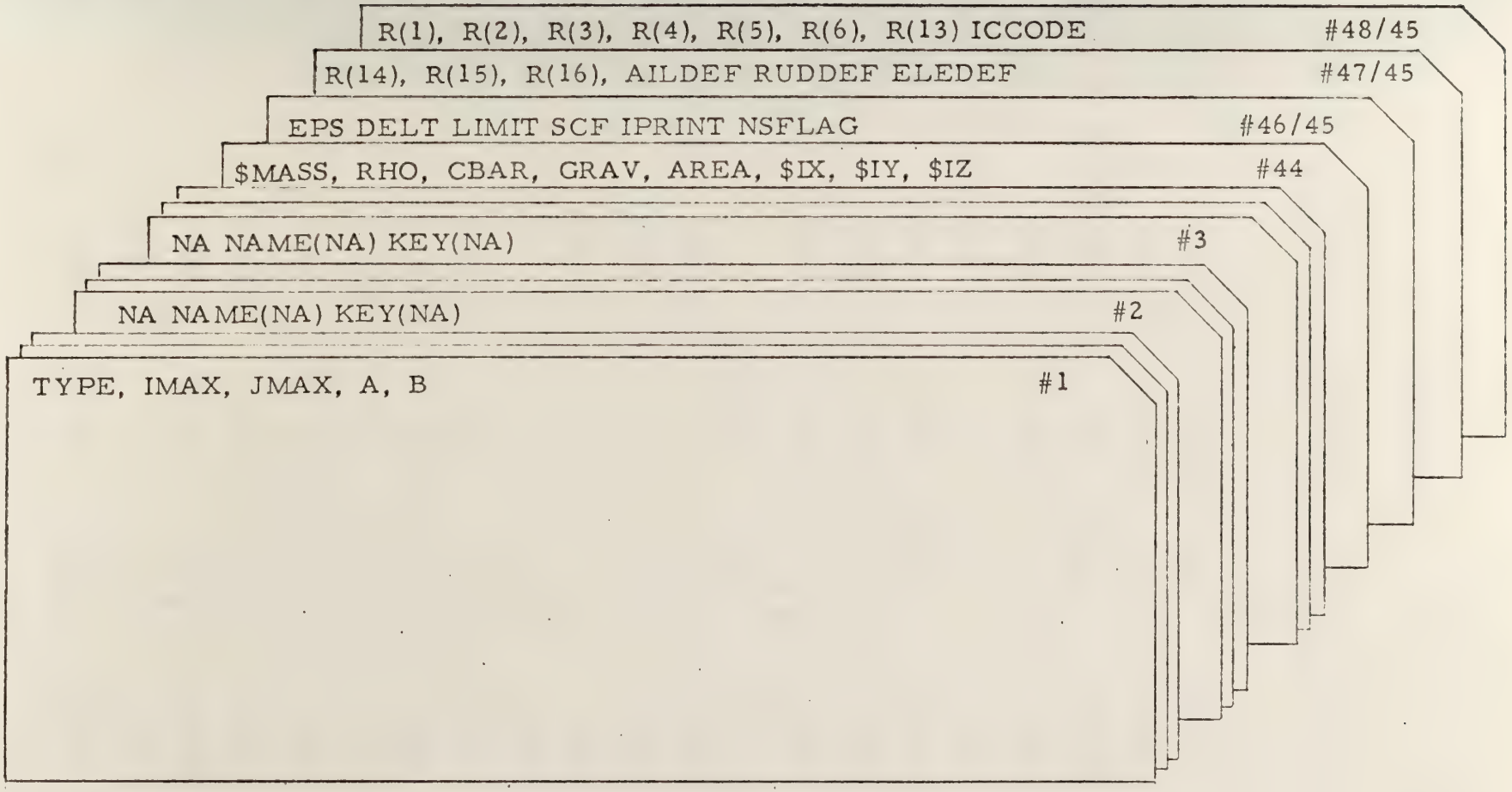



Figure B-1

Data Deck Composition for the Airplane Spin Solution Program

TABLE B-I

Force and Moment Coefficients, Stability Derivatives, and Associated 'NA' Assignments

| | | | | | |
|---|---|---|---|---|---|
| 1 | $C_x$ | CX | 22 | $C_{n_{\delta r}}$ | CNDR |
| 2 | $C_y$ | XY | 23 | $C_{x_r}$ | CXR |
| 3 | $C_z$ | CZ | 24 | $C_{x_{\delta r}}$ | CXDR |
| 4 | $C_l$ | CL | 25 | $C_{x_p}$ | CXP |
| 5 | $C_m$ | CM | 26 | $C_{x_q}$ | CXQ |
| 6 | $C_n$ | CN | 27 | $C_{x_{\delta a}}$ | CXDA |
| 7 | $C_{x_{\delta e}}$ | CZDE | 28 | $C_{y_q}$ | CYQ |
| 8 | $C_{y_p}$ | CYP | 29 | $C_{y_{\delta e}}$ | CYDH |
| 9 | $C_{y_r}$ | CYR | 30 | $C_{z_p}$ | CZP |
| 10 | $C_{y_{\delta a}}$ | CYDA | 31 | $C_{z_q}$ | CZQ |
| 11 | $C_{y_{\delta r}}$ | CYDR | 32 | $C_{z_r}$ | CZR |
| 12 | $C_{z_{\delta e}}$ | CZDE | 33 | $C_{z_{\delta a}}$ | CZDA |
| 13 | $C_{l_p}$ | CLP | 34 | $C_{z_{\delta r}}$ | CZDR |
| 14 | $C_{l_r}$ | CLR | 35 | $C_{l_q}$ | CLQ |
| 15 | $C_{l_{\delta a}}$ | CLDA | 36 | $C_{l_{\delta e}}$ | CLDE |
| 16 | $C_{l_{\delta r}}$ | CLDR | 37 | $C_{m_p}$ | CMP |
| 17 | $C_{m_q}$ | CMQ | 38 | $C_{m_r}$ | CMR |
| 18 | $C_{m_{\delta e}}$ | CMDE | 39 | $C_{m_{\delta a}}$ | CMDA |
| 19 | $C_{n_p}$ | CNP | 40 | $C_{m_{\delta r}}$ | CMDR |
| 20 | $C_{n_r}$ | CNR | 41 | $C_{n_q}$ | CNQ |
| 21 | $C_{n_{\delta a}}$ | CNDA | 42 | $C_{n_{\delta e}}$ | CNDE |

conditions included as data set #48 need be provided. This option is effected by assigning NSFLAG = 2 or 3 as explained in the program comments listing and Table B-II.

It is suggested that the user who wishes to test a large number of initial conditions assign LIMIT = 10, NSFLAG = 2 and then test all of the initial conditions on one job. This scheme will quickly reveal to the user which initial conditions exceed the alpha/beta limits without wasting valuable computation time. Subsequently, those initial conditions which appear to have a solution potential can be submitted with appropriate data set #45 values.

Since every iteration generates a new set of initial conditions, the program can be restarted merely by including as data set #45 the appropriate values taken from the printed output. This option allows the user to check the stability of a quasi-steady spin solution by changing the time increment (DELT), one of the solution convergence factors (SCF), the control deflections (AILDEF, RUDDEF, ELEDEF), or the atmospheric density (RHO).

## C. DATA OUTPUT

All output is formated for 130 character computer paper. An option is provided by NPRINT to suppress the iterative results; otherwise, all input data is echo checked, all significant iterative computations are printed and a summary of the integration solution path is provided.

## D. STORAGE REQUIREMENTS

The amount of storage required is primarily a function of the size of the CFORCM three dimensional matrix. The third index of the matrix is associated with, and is no larger than, the largest NA

TABLE B-II

NSFLAG Integer Assignment

| NSFLAG | SOLUTION METHOD | EQUATION FORM | SUBSEQUENT DATA SET COMPOSITION |
|---|---|---|---|
| 0 | Integration | Modified | #45 |
| 1 | Integration | Full | #45 |
| 2 | Integration | Modified | #48 |
| 3 | Integration | Full | #48 |
| 4 | Gradient | Short | #45 |

integer (NSMAX) corresponding to vector or matrix coefficient data sets. The first and second indices are associated with IMAX and JMAX. Prior to using the program the user should dimension A, B, and CFORCM in the COMMON/WORKA/ specification statements to A(IMAX), B(JMAX), and CFORCM(IMAX, JMAX, NAMAX). The WORKA statement must be identical in the following routines: MAIN, GAWAIN, SOLVIT, EQNS, INTER, and ANGLE. Common regions are utilized for most communications between subroutines and therefore care must be exercised to ensure that the common statements are identical.

## E. PROGRAM FLOWCHARTS

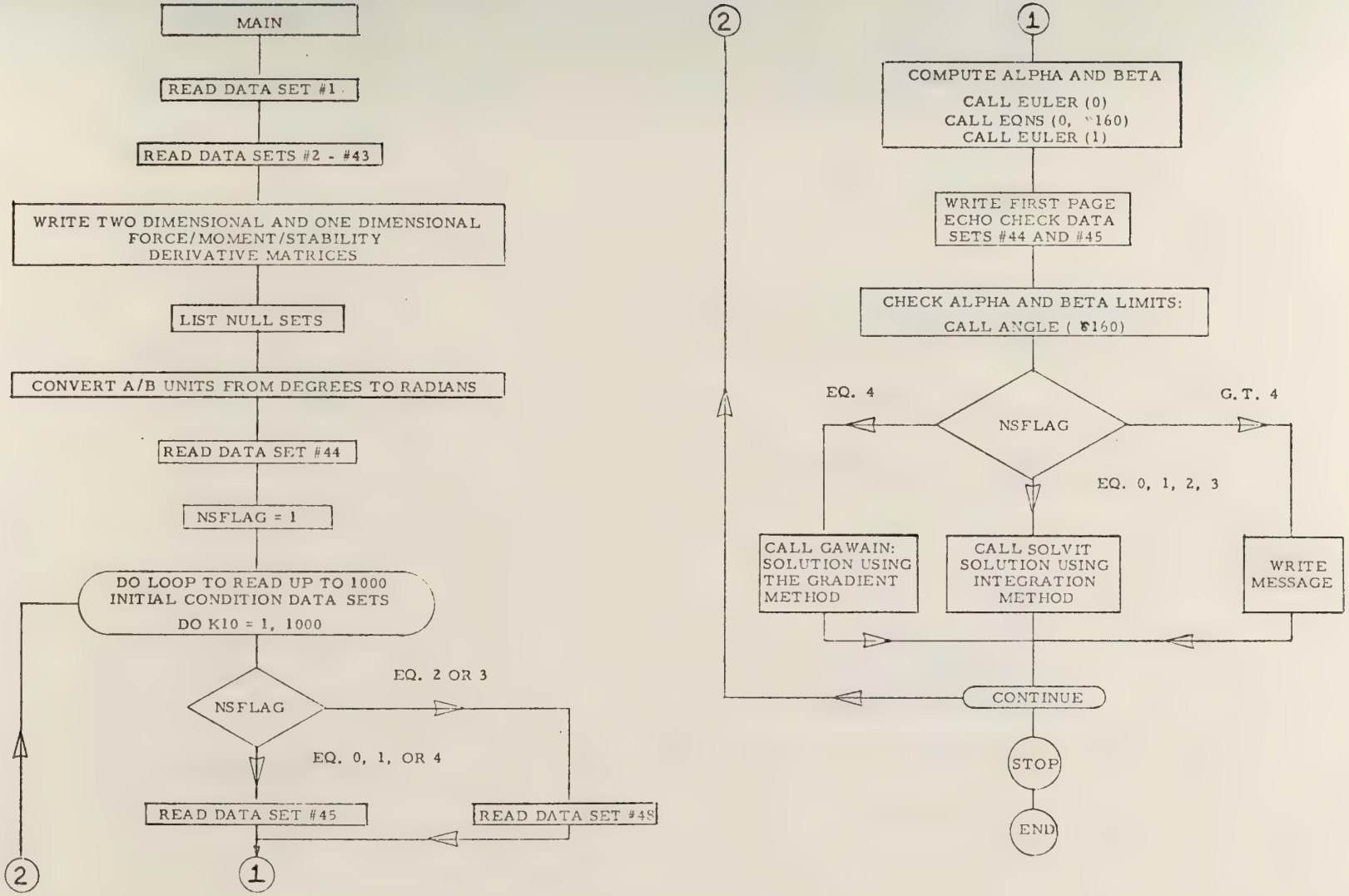
Main and subroutine flow charts are included as Figures B-2 to B-9 and are intended to be self-explanatory. The program source listing includes a sufficient number of comments to amplify the flowchart information. Subroutines SPLIN1 and SPLICO were provided by the NPS Computer Center. No effort has been made to provide reference flowcharts as both routines have been extensively tested and validated by the NPS staff.

## F. PROGRAMMING CONSIDERATIONS

One should note that calls on Euler are made in pairs. The angular variables are stored in the common memory, thus if the first conversion is from equation compatible to output variables, the second call must effect the opposite conversion.

Subroutine GAWAIN requires the computation of the partial derivative of the criterion with respect to a change in each independent variable. This is accomplished by incrementing a single independent variable by

FAC. The criterion is then computed using this value. The change in the criterion function value is then divided by FAC to obtain the partial derivative.




MAIN Program Flowchart

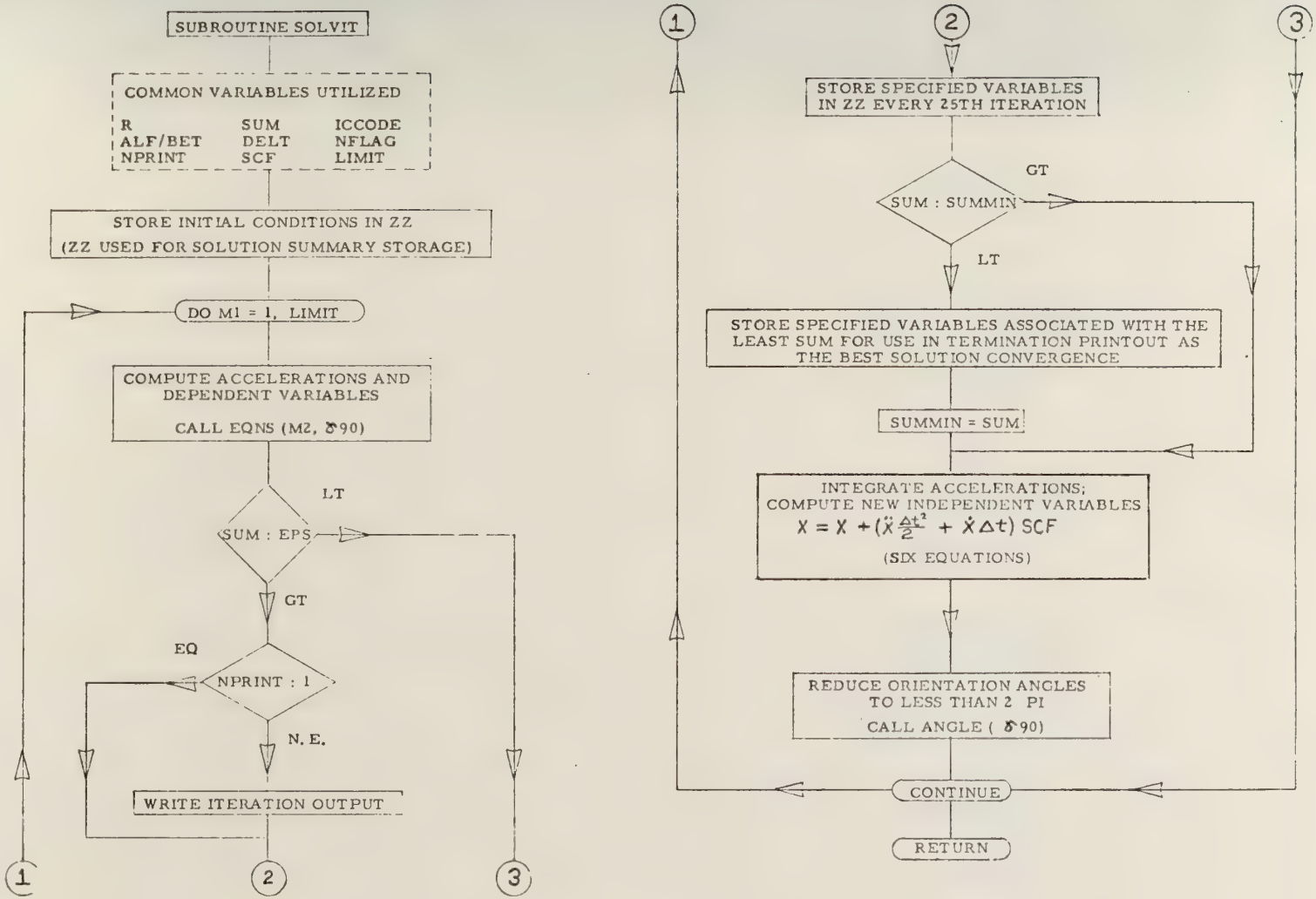



Figure B-3

SOLVIT Subroutine Flowchart

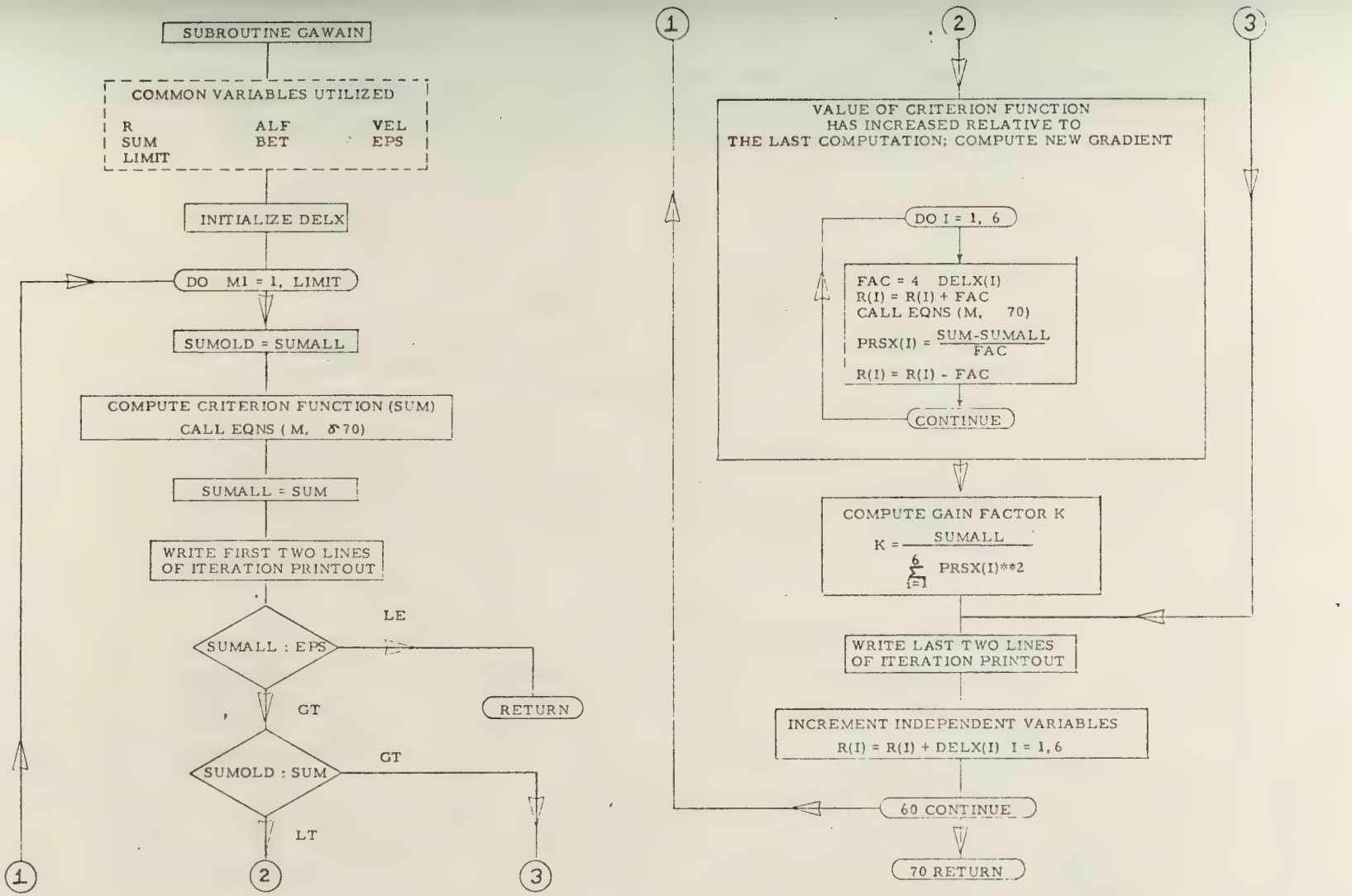



Figure B-4

GAWAIN Subroutine Flowchart

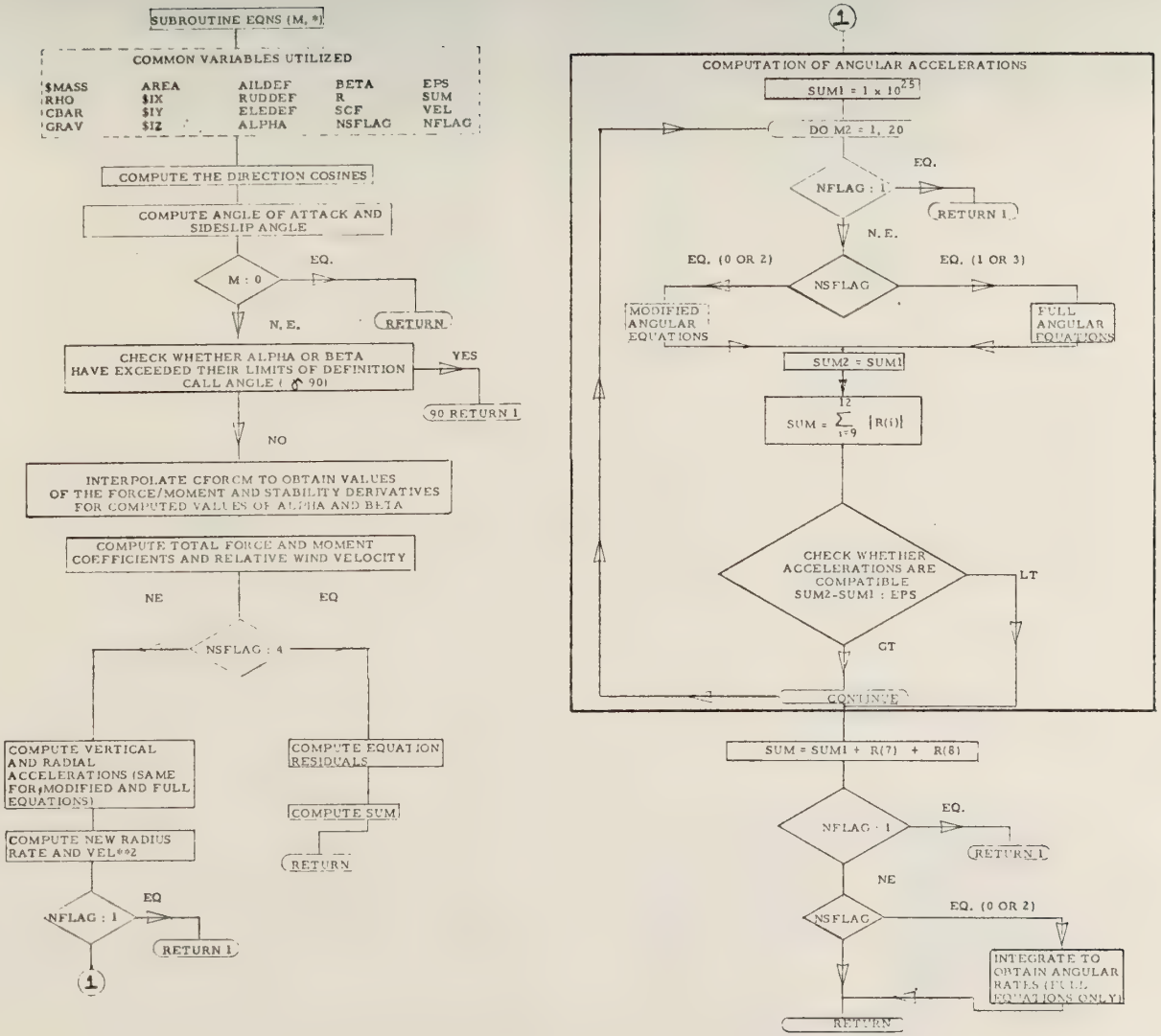



Figure B-5

EQNS Subroutine Flowchart

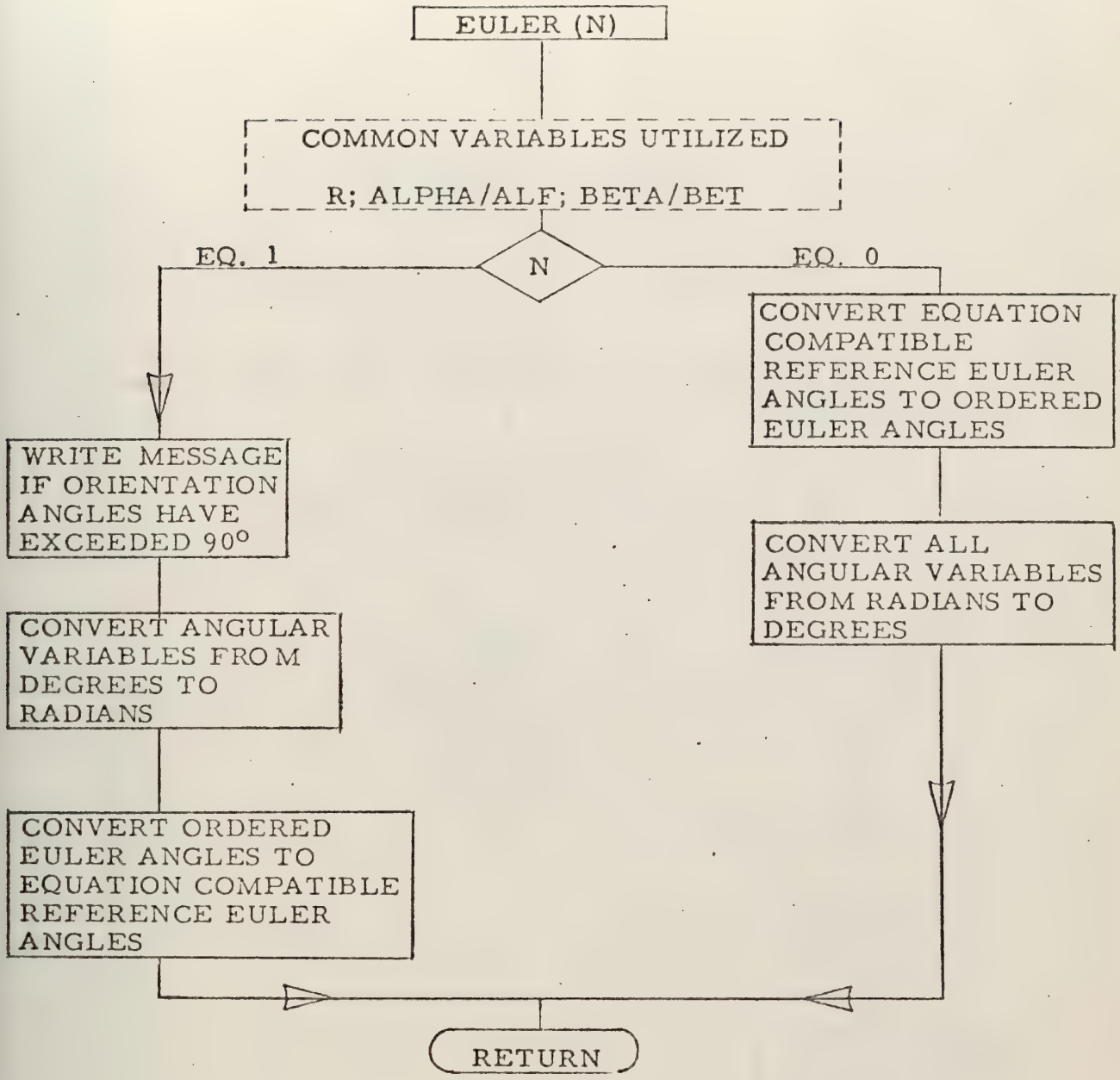

EULER (N)
COMMON VARIABLES UTILIZED
R; ALPHA/ALF; BETA/BET
N
EQ. 1
EQ. 0
WRITE MESSAGE IF ORIENTATION ANGLES HAVE EXCEEDED 90°
CONVERT ANGULAR VARIABLES FROM DEGREES TO RADIANS
CONVERT ORDERED EULER ANGLES TO EQUATION COMPATIBLE REFERENCE EULER ANGLES
CONVERT EQUATION COMPATIBLE REFERENCE EULER ANGLES TO ORDERED EULER ANGLES
CONVERT ALL ANGULAR VARIABLES FROM RADIANS TO DEGREES
RETURN

Figure B-6

EULER Subroutine Flowchart

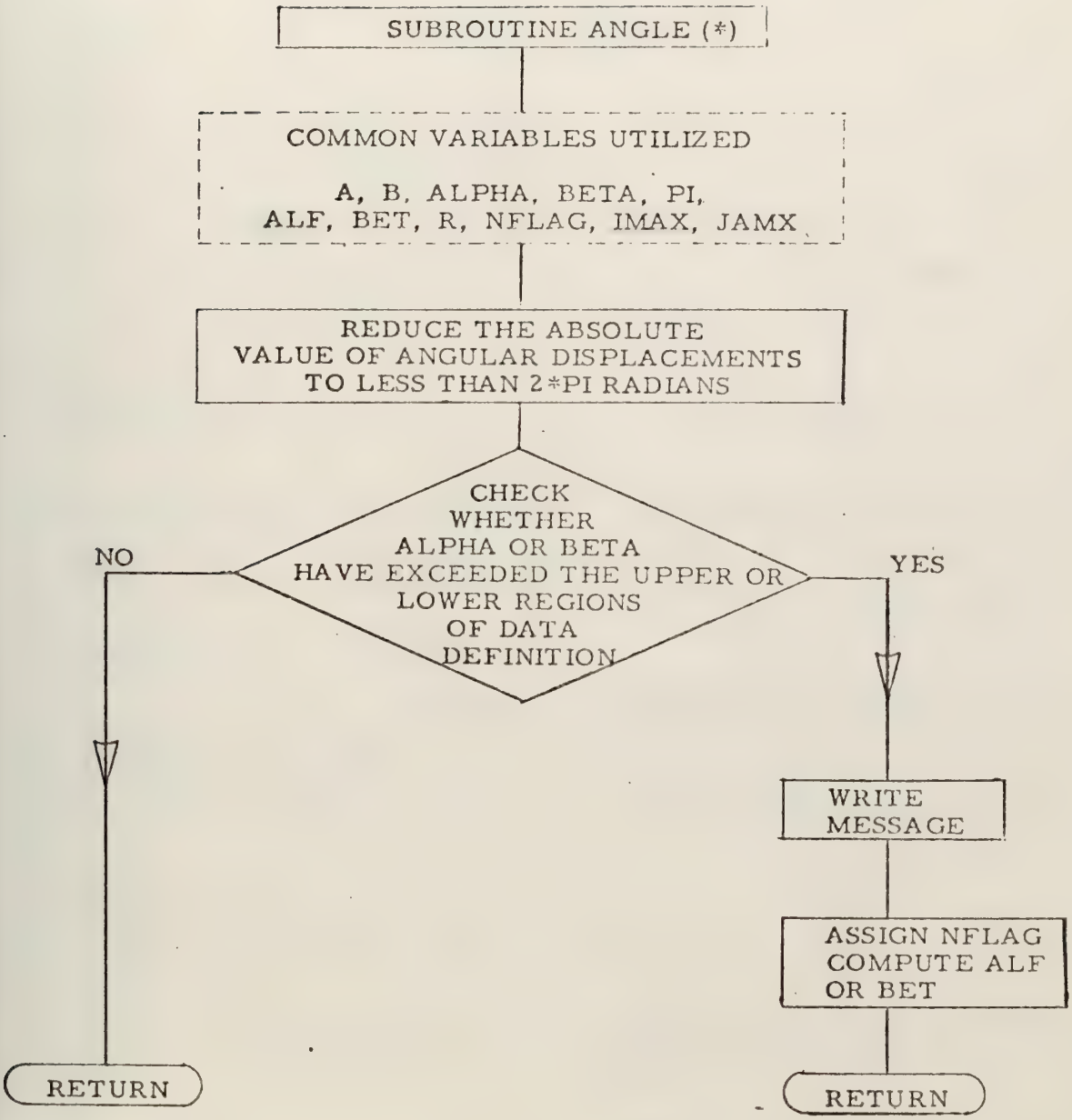



Figure B-7

ANGLE Subroutine Flowchart

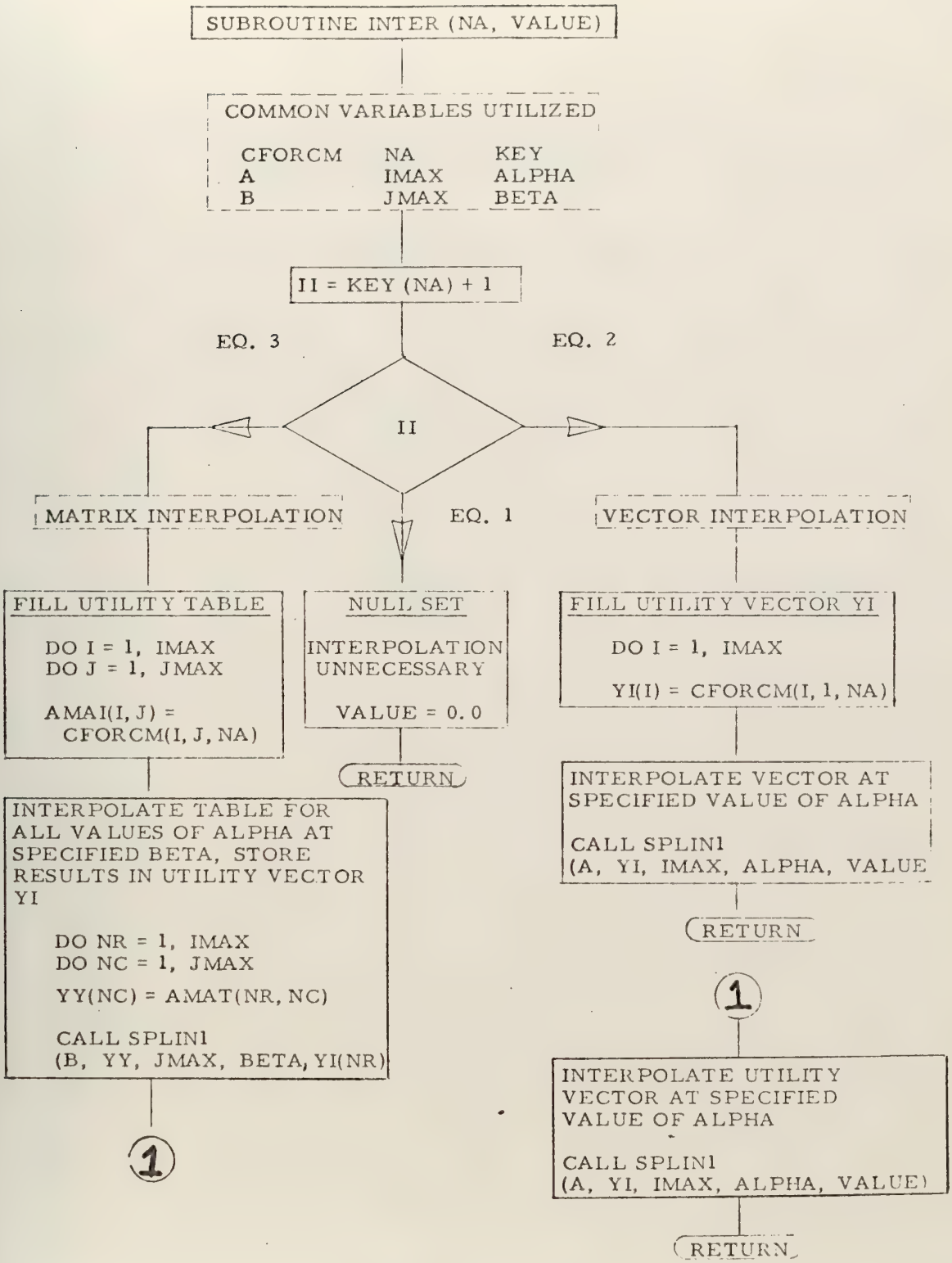



Figure B-8

INTER Subroutine Flowchart

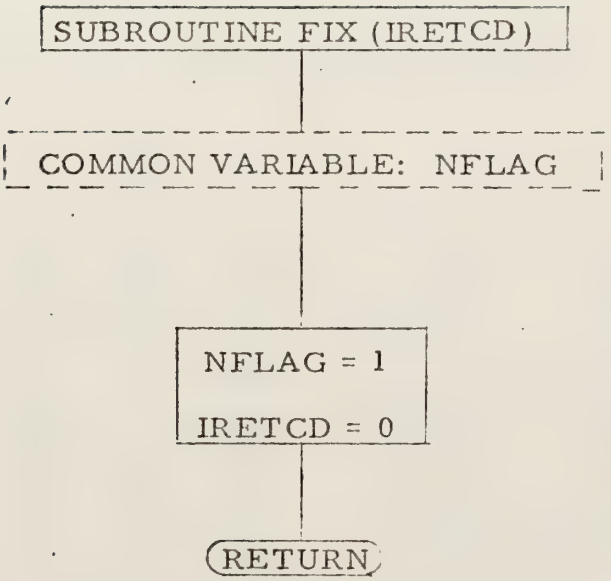



Figure B-9

FIX Subroutine Flowchart

```
C
C     AIRCRAFT SPIN SOLUTION PROGRAM                                    * MAI00010
C                                                                       * MAI00020
C     GENERAL INFORMATION                                               * MAI00030
C          THIS PROGRAM SOLVES AIRPLANE EQUATIONS OF MOTION BY TWO      * MAI00040
C          METHODS. THE EQUATIONS UTILIZED ARE MODELED IN A CYLIN-      * MAI00050
C          DRICAL COORDINATE SYSTEM IN ORDER TO OBVIATE A STEADY        * MAI00060
C          OR QUASI STEADY STATE SOLUTION FOR A SPINNING AIRCRAFT.      * MAI00070
C                                                                       * MAI00080
C     EQUATION FORMS                                                    * MAI00090
C          FULL:  SIX EXTENSIVELY COUPLED, HIGHLY NON LINEAR            * MAI00100
C          DIFFERENTIAL EQUATIONS. INCLUDED ARE ALL APPLICABLE          * MAI00110
C          ZERO, FIRST, AND SECOND ORDER DERIVATIVES OF THE INDE-       * MAI00120
C          PENDENT VARIABLES.                                           * MAI00130
C                                                                       * MAI00140
C          MODIFIED:  SIMILAR TO THE FULL EQUATIONS, HOWEVER THE        * MAI00150
C          DERIVATIVES OF THE ORIENTATION VARIABLES ARE EQUATED         * MAI00160
C          TO ZERO.                                                     * MAI00170
C                                                                       * MAI00180
C          SHORT:  SIX ALGEBRAIC EQUATIONS WHICH ARE DERIVED FROM       * MAI00190
C          THE FULL EQUATIONS BY EQUATING ALL APPLICABLE FIRST          * MAI00200
C          AND SECOND ORDER DERIVATES TO ZERO.                          * MAI00210
C                                                                       * MAI00220
C     SOLUTION METHODS                                                  * MAI00230
C          INTEGRATION METHOD:  ITERATIVE SOLUTION WHEREBY THE          * MAI00240
C          INDEPENDENT VARIABLE ACCELERATIONS ARE COMPUTED AS A         * MAI00250
C          FUNCTION OF THE INPUTTED INITIAL CONDITIONS. NEW INI-        * MAI00260
C          TIAL CONDITIONS ARE THEN COMPUTED BY INTEGRATING THE         * MAI00270
C          ACCELERATIONS. A SOLUTION IS OBTAINED WHEN THE ACCEL-        * MAI00280
C          ERATIONS REACH A MINIMUM VALUE. THIS METHOD IS APPLI-        * MAI00290
C          CABLE ONLY TO THE FULL AND MODIFIED EQUATIONS.               * MAI00300
C                                                                       * MAI00310
C          GRADIENT METHOD:  ITERATIVE SOLUTION WHEREBY A CRITE-        * MAI00320
C          RION FUNCTION IS COMPUTED FROM INPUTTED INITIAL CONDI-       * MAI00330
C          TIONS; THE GRADIENT TO THE FUNCTION IS USED TO GENERATE      * MAI00340
C          NEW INITIAL CONDITIONS. A STEADY STATE SOLUTION IS           * MAI00350
C          ACHIEVED WHEN THE CRITERION FUNCTION EQUALS ZERO. THIS       * MAI00360
C          METHOD IS APPLICABLE TO THE SHORT EQUATIONS ONLY.            * MAI00370
C                                                                       * MAI00380
C     DESCRIPTION OF PARAMETERS (AIRCRAFT/ENVIRONMENTAL CONSTANTS)      * MAI00390
C          $MASS:          MASS OF AIRCRAFT, SLUGS                      * MAI00400
C          CBAR:           MEAN AERODYNAMIC CHORD, FEET                 * MAI00410
C          AREA:           WING AREA, FEET**2                           * MAI00420
C          $IX,$IY,$IZ:    PRINCIPAL MOMENTS OF INERTIA,                * MAI00430
C                          SLUG-FEET**2                                 * MAI00440
C          TYPE:           AIRCRAFT TYPE, I.E., F-4, F111               * MAI00450
```

```
C              NA:             INTEGER CORRESPONDING TO A PARTICULAR    * MAI00460
C                              STABILITY DERIVATIVE. (REFER TO THE FIRST* MAI00470
C                              REFERENCE FOR THE INTEGER ASSIGNMENT)    * MAI00480
C              KEY(42):        VECTOR OF INTEGERS WHICH INDICATE FORM OF* MAI00490
C                              STABILITY DERIVATIVE                     * MAI00500
C                              KEY = 0:  NULL SET                       * MAI00510
C                              KEY = 1:  1-D VECTOR THAT IS A FUNCTION  * MAI00520
C                              ONLY OF ALPHA                            * MAI00530
C                              KEY = 2:  2-D MATRIX THAT IS A FUNCTION  * MAI00540
C                              OF BOTH ALPHA AND BETA                   * MAI00550
C              NAME(42):       VECTOR OF LITERAL NAMES OF THE STABILITY * MAI00560
C                              DERIVATIVES                              * MAI00570
C              RHO:            AIR DENSITY, SLUG/FEET**3                * MAI00580
C              GRAV:           GRAVITATIONAL CONSTANTS, FEET/SECOND**2  * MAI00590
C              CFORCM:         THREE DIMENSIONAL DATA BANK CONTAINING   * MAI00600
C                              THE STABILITY DERIVATIVE MATRICES.       * MAI00610
C                              DIMENSIONED TO AT LEAST (IMAX,JMAX,NAMAX)* MAI00620
C                              WHERE NAMAX IS EQUAL TO THE LARGEST INTE-* MAI00630
C                              GER CONTAINED IN KEY                     * MAI00640
C              A(IMAX):        VECTOR OF THE VALUES OF AIRCRAFT ANGLE OF* MAI00650
C                              ATTACK FOR WHICH THERE IS A ONE-FOR-ONE   * MAI00660
C                              CORRESPONDENCE TO THE ORDINATE VALUES    * MAI00670
C                              OF THE STABILITY DERIVATIVE MATRICES,    * MAI00680
C                              DEGREES                                  * MAI00690
C              B(JMAX):        VECTOR OF THE VALUES OF AIRCRAFT SIDESLIP* MAI00700
C                              ANGLE FOR WHICH THERE IS A ONE-FOR-ONE   * MAI00710
C                              CORRESPONDENCE TO THE ABSCISSA VALUES    * MAI00720
C                              OF THE STABILITY DERIVATIVE MATRICES,    * MAI00730
C                              DEGREES                                  * MAI00740
C              IMAX/JMAX:      INTEGER EQUAL TO THE NUMBER OF VALUES    * MAI00750
C                              IN THE A/B VECTOR                        * MAI00760
C              ALPHA/ALF:      COMPUTED ANGLE OF ATTACK, RADIANS/DEGREES* MAI00770
C              BETA/BET:       COMPUTED SIDESLIP ANGLE, RADIANS/DEGREES * MAI00780
C                                                                       * MAI00790
C         DESCRIPTION OF PARAMETERS (SOLUTION VARIABLES)                * MAI00800
C              DELT:           INTEGRATION TIME INCREMENT, SECONDS      * MAI00810
C              SCF(6):         SOLUTION CONVERGENCE FACTORS             * MAI00820
C              LIMIT:          ITERATION DO LOOP LIMIT                  * MAI00830
C              NFLAG:          INTEGER ASSOCIATED WITH THE EXTENDED     * MAI00840
C                              ERROR HANDLING FACILITY.(REFER TO SUBROU-* MAI00850
C                              TINE 'FIX') ALSO USED TO COMMUNICATE     * MAI00860
C                              ABNORMAL TERMINATION INFORMATION.          MAI00870
C              NSFLAG:         INTEGER WHICH DETERMINES THE SOLUTION    * MAI00880
C                              METHOD AND FORM OF EQUATIONS TO BE       * MAI00890
C                              UTILIZED                                 * MAI00900
C              NPRINT:         INTEGER WHICH IF EQUAL TO ONE WILL SUP-  * MAI00910
C                              PRESS THE ITERATION OUTPUT BUT WILL      * MAI00920
C                              PROVIDE A SUMMARY OF THE SOLUTION        * MAI00930
```

```
C          ICCODE:       SIX DIGIT INTEGER SERIAL NUMBER OF INPUT  * MAI00940
C                        INITIAL CONDITONS                         * MAI00950
C          SUM:          SUM OF THE ABSOLUTE VALUES OF THE INDE-   * MAI00960
C                        PENDENT VARIABLE ACCELERATIONS            * MAI00970
C          EPS:          REAL VALUE USED TO DETERMINE CONVERGENCE  * MAI00980
C          AILDEF:       AIRCRAFT AILERON DEFLECTION, DEGREES      * MAI00990
C                        RIGHT AILERON TRAILING EDGE DOWN IS POS-  * MAI01000
C                        TIVE                                      * MAI01010
C          RUDDEF:       AIRCRAFT RUDDER DEFLECTION, DEGREES       * MAI01020
C                        TRAILING EDGE LEFT IS POSITIVE            * MAI01030
C          ELEDEF:       AIRCRAFT ELEVATOR OR STABILATOR DEFLEC-   * MAI01040
C                        TION, DEGREES, TRAILING EDGE DOWN IS      * MAI01050
C                        POSITIVE.                                 * MAI01060
C                                                                  * MAI01070
C     DESCRIPTION OF PARAMETERS (INDEPENDENT VARIABLES)            * MAI01080
C          R:            VECTOR OF THE INDEPENDENT VARIABLES AND   * MAI01090
C                        ASSOCIATED DERIVATIVES AS DESCRIBED BELOW* MAI01100
C                                                                  * MAI01110
C          R(1):         ALTITUDE RATE, FEET/SEC, NEGATIVE DIREC-  * MAI01120
C                        TION IS DOWNWARDS                         * MAI01130
C          R(2):         RADIUS, (FEET)                            * MAI01140
C          R(3):         SPINRATE, DEGREES/SECOND                  * MAI01150
C          R(4):         THETA, PITCH EULER ANGLE, DEGREES         * MAI01160
C          R(5):         PSI, ROLL EULER ANGLE, DEGREES            * MAI01170
C          R(6):         PHI, YAW EULER ANGLE, DEGREES             * MAI01180
C          R(7):         VERTICAL ACCELERATION, FEET/SECOND**2     * MAI01190
C          R(8):         RADIUS ACCELERATION, FEET/SECOND**2       * MAI01200
C          R(9):         ANGULAR ACCELERATION, RADIANS/SECOND**2   * MAI01210
C          R(10):        PITCH ACCELERATION, RADIANS/SECOND**2     * MAI01220
C          R(11):        ROLL ACCELERATION, RADIANS/SECOND**2      * MAI01230
C          R(12):        YAW ACCELERATION, RADIANS/SECOND**2       * MAI01240
C          R(13):        RADIUS RATE, FEET/SECOND                  * MAI01250
C          R(14):        PITCH RATE, RADIANS/SECOND                * MAI01260
C          R(15):        ROLL RATE, RADIANS/SECOND                 * MAI01270
C          R(16):        YAW RATE, RADIANS/SECOND                  * MAI01280
C                                                                  * MAI01290
C          NOTE:         ORIENTATION ORDER IS  PITCH, ROLL AND     * MAI01300
C                        THEN YAW                                  * MAI01310
C                                                                  * MAI01320
C     USAGE                                                        * MAI01330
C          THE PROGRAM READS THE STABILITY DERIVATIVE MATRICES     * MAI01340
C          AND RELATED AIRCRAFT CONSTANTS AND THEN COMMENCES TO    * MAI01350
C          ITERATIVELY SEEK A SOLUTION BASED UPON THE FIRST INI-   * MAI01360
C          TIAL CONDITION DATA SET THAT IS READ.  UPON TERMINATION* MAI01370
C          OF A PARTICULAR SOLUTION, THE PROGRAM THEN READS SUBSE-* MAI01380
C          QUENT INITIAL CONDITION DATA SETS; THE FORMAT OF WHICH  * MAI01390
C          IS EXPLAINED BELOW UNDER 'DATA DECK COMPOSITION'        * MAI01400
C                                                                  * MAI01410
```

```
C
C          SUBROUTINES REQUIRED/PURPOSE                                    * MAI01420
C               SOLVIT:        INTEGRATION SOLUTION                        * MAI01430
C               GAWAIN:        GRADIENT SOLUTION                           * MAI01440
C               EQNS:          EQUATIONS (THREE FORMS)                     * MAI01450
C               ANGLE:         CHECKS ALPHA AND BETA LIMITS                * MAI01460
C               EULER:         CONVERTS EULER ANGLES                       * MAI01470
C               INTER:         INTERPOLATION CONTROL                       * MAI01480
C               SPLIN:         CUBIC INTERPOLATION                         * MAI01490
C               ERRSET         OVERRIDES FORTRAN ERROR HANDLING FACILITY   * MAI01500
C               FIX            FACILITATES EXECUTION TERMINATION           * MAI01510
C                                                                          * MAI01520
C          SOLUTION METHOD AND EQUATION FORM                               * MAI01530
C               NSFLAG = 0 OR 2:  MODIFIED EQUATIONS/INTEGRATION METHOD    * MAI01540
C               NSFLAG = 1 OR 3:  FULL EQUATIONS/INTEGRATION METHOD        * MAI01550
C               NSFLAG = 4:       SHORT EQUATIONS/GRADIENT METHOD          * MAI01560
C                                                                          * MAI01570
C          PRINTED OUTPUT                                                  * MAI01580
C               THE AMOUNT OF PRINTED OUTPUT IS DETERMINED BY NPRINT.      * MAI01590
C               THE DEFAULT VALUE OF (0) PROVIDES AN ECHO CHECK OF ALL     * MAI01600
C               INITIAL CONDITIONS AND SOLUTION PARAMETERS, A PRINTOUT     * MAI01610
C               OF EACH SOLUTION ITERATION AND, AFTER TERMINATION, A       * MAI01620
C               SUMMARY OF THE SOLUTION PATH IS PROVIDED                   * MAI01630
C                                                                          * MAI01640
C          JOB CONTROL LANGUAGE (EXECUTION, USING SOURCE DECK)             * MAI01650
C               1. //JOB                                                   * MAI01660
C               2. // EXEC FORTCLG,REGION.GO=135K  (FORTRAN IV G)          * MAI01670
C               OR // EXEC FORTHCLG,PARM.FORT='OPT=2',REGION.GO=135K       * MAI01680
C                  (FORTRN IV H)                                           * MAI01690
C               3. SOURCE DECK                                             * MAI01700
C               4. /*                                                      * MAI01710
C               5. //GO.FT06F001 DD SYSOUT=A,SPACE=(CYL(3,2))             * MAI01720
C               6. //GO.SYSIN DD *                                         * MAI01730
C               7. DATA DECK                                               * MAI01740
C               8. /*                                                      * MAI01750
C                                                                          * MAI01760
C          REFERENCES                                                      * MAI01770
C               CHAMPOUX, R.L. 'THE NUMERICAL SOLUTION AND ANALYSIS OF     * MAI01780
C               AIRPLANE SPIN EQUATIONS MODELED IN AN INERTIAL COORDI-     * MAI01790
C               NATE SYSTEM', MASTERS THESIS, UNITED STATES NAVAL POST-    * MAI01800
C               GRADUATE SCHOOL, MONTEREY, CALIFORNIA, DECEMBER 1972       * MAI01810
C                                                                          * MAI01820
C               BUEHLER, R.R. 'THE DERIVATION, SOLUTION, AND ANALYSIS      * MAI01830
C               OF AIRPLANE SPIN EQUATIONS MODELED IN AN INERTIAL COOR-     * MAI01840
C               DINATE SYSTEM', MASTERS THESIS, UNITED STATES NAVAL        * MAI01850
C               POST GRADUATE SCHOOL, MONTEREY, CALIFORNIA, MARCH 1972     * MAI01860
C                                                                          * MAI01870
C                                                                          * MAI01880
C                                                                          * MAI01890
```

```
C                          DATA DECK COMPOSITION                              * MAI01900
C*************************************************************************** MAI01910
C DATA SET *                VARIABLES                    *       FORMAT       * MAI01920
C*************************************************************************** MAI01930
C     1      * TYPE,IMAX,JMAX,(A(I),I=1,IMAX),          * (A4,2I3,7F10.2,     * MAI01940
C            *    (B(J),J=1,JMAX)                       *   (8F10.2))         * MAI01950
C*************************************************************************** MAI01960
C            * NA, NAME(NA), KEY(NA)  THE REMAINDER*  (I10,10X,A4,10X,I6)* MAI01970
C            *   EACH DATA SET IS DETERMINED BY KEY*                         * MAI01980
C            ************************************************************** MAI01990
C            *              IF(KEY.EQ.2)                                     * MAI02000
C            * ((CFORCM(I,J,NA),J=1,JMAX),I=1,IMAX)*   (8F10.6)             * MAI02010
C   2-43     ************************************************************** MAI02020
C(42 SETS)   *              IF(KEY.EQ.1)                                     * MAI02030
C            *  (CFORCM(I,1,NA),I=1,IMAX)           *   (8F10.6)            * MAI02040
C            ************************************************************** MAI02050
C            *  IF(KEY.EQ.0) THE NA STABILITY       *                       * MAI02060
C            * DERIVATIVE MATRIX IS NULL AND NO     * (NOT APPLICABLE)      * MAI02070
C            * DATA CARDS WILL BE READ              *                       * MAI02080
C*************************************************************************** MAI02090
C    44      * $MASS,RHO, CBAR, GRAV, AREA, $IX     *   (8F10.6)            * MAI02100
C            *  $IY, $IZ                            *                       * MAI02110
C*************************************************************************** MAI02120
C     * 46   * EPS, DELT, LIMIT, (SCF(I),I=1,6),    * (2F10.6,I10,6F5.2,    * MAI02130
C     *      *  NPRINT, NSFLAG                      *   2I10)               * MAI02140
C     ************************************************************** MAI02150
C     * 47   * R(14), R(15), R(16), AILDEF, RUDDEF,*   (8F10.6)             * MAI02160
C 45  *      *  ELEDE=                              *                       * MAI02170
C     ************************************************************** MAI02180
C     * 48   * (R(I),I=1,6), R(13), ICCODE          *   (7F10.6,I10)        * MAI02190
C*************************************************************************** MAI02200
C                                                                            * MAI02210
C                NOTE:     THE FIRST SET OF INITIAL CONDITIONS MUST BE     * MAI02220
C                          PROVIDED AS DATA SET #45.  THE COMPOSITION      * MAI02230
C                          OF THE SUBSEQUENT SETS OF INITIAL CONDITIONS    * MAI02240
C                          IS DETERMINED BY NSFLAG                         * MAI02250
C                                                                          * MAI02260
C                          IF NSFLAG= 0,1,OR 4, THE USER MUST PROVIDE      * MAI02270
C                          DATA SET #45 AS THE NEXT SET OF INITIAL         * MAI02280
C                          CONDITIONS                                      * MAI02290
C                                                                          * MAI02300
C                          IF NSFLAG = 2 OR 3, THE VARIABLES IN DATA       * MAI02310
C                          SETS #46 AND #47 WILL REMAIN CONSTANT FOR ALL   * MAI02320
C                          SUBSEQUENT COMPUTATIONS. THE USER MAY ONLY      * MAI02330
C                          PROVIDE SUBSEQUENT INITIAL CONDITIONS AS DATA   * MAI02340
C                          SET #48 AND ALL COMPUTATIONS WILL BE MADE       * MAI02350
C                          USING THE INTEGRATION METHOD                    * MAI02360
C                                                                          * MAI02370
```

```
      IMPLICIT REAL *8 (A-H,O-Z,$)                                          MAI02380
      COMMON NFLAG,NPRINT,ICCODE                                            MAI02390
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,       MAI02400
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF               MAI02410
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,       MAI02420
     1EPS,SUM ,VEL                                                          MAI02430
      REAL NAME(42),TYPE                                                    MAI02440
      INTEGER P(9),A1,A2                                                    MAI02450
      DIMENSION N1(42),N2(42),NO(42)                                        MAI02460
      CALL ERRSET(207,100,-1,1,FIX,209)                                     MAI02470
      NOCNTR=0                                                              MAI02480
      N1CNTR=0                                                              MAI02490
      N2CNTR=0                                                              MAI02500
C                                                                           *MAI02510
CCC       READ DATA SET #1                                                  *MAI02520
CC                                                                          *MAI02530
      READ(5,1000) TYPE,IMAX,JMAX,(A(II),II=1,IMAX),(B(J),J=1,JMAX)         MAI02540
C                                                                           *MAI02550
CCC       READ DATA SETS #2 - #43                                           *MAI02560
C                                                                           *MAI02570
      DO   30 K1=1,42                                                       MAI02580
      READ (5,1001)NA,NAME(NA),KEY(NA)                                      MAI02590
      IF(KEY(NA).EQ.1)READ(5,1002)(CFORCM(I,1,NA),I=1,IMAX)                 MAI02600
      IF(KEY(NA).EQ.2)READ(5,1002)((CFORCM(I,J,NA),J=1,JMAX),I=1,IMAX)      MAI02610
      IF(KEY(NA).EQ.0)GO TO 20                                              MAI02620
      IF(KEY(NA).EQ.2) GO TO   10                                           MAI02630
C                                                                           *MAI02640
CCC       COUNT THE NUMBER OF STABILITY DERIVATIVE MATRICES WHICH           *MAI02650
CCC       HAVE KEY=0/1/2; PLACE NA VALUES IN THE APPROPRIATE QUEUES         *MAI02660
CCC       (NO, N1, N2)                                                      *MAI02670
C                                                                           *MAI02680
      N1CNTR=N1CNTR+1                                                       MAI02690
      N1(N1CNTR)=NA                                                         MAI02700
      GO TO   30                                                            MAI02710
   10 N2CNTR=N2CNTR+1                                                       MAI02720
      N2(N2CNTR)=NA                                                         MAI02730
      GO TO   30                                                            MAI02740
   20 NOCNTR=NOCNTR+1                                                       MAI02750
      NO(NOCNTR)=NA                                                         MAI02760
   30 CONTINUE                                                              MAI02770
      IF(NPRINT.EQ.1) GO TO 120                                             MAI02780
      NOMAX=NOCNTR                                                          MAI02790
      N1MAX=N1CNTR                                                          MAI02800
      N2MAX=N2CNTR                                                          MAI02810
C                                                                           *MAI02820
CCC       WRITE THE TWO DIMENSIONAL MATRICES  (N2 QUEUE)                    *MAI02830
C                                                                           *MAI02840
      DO   60 K2=1,N2MAX                                                    MAI02850
```

```
      DO 40 J=1,JMAX                                                    MAI02860
   40 P(J)=IDINT(B(J))                                                  MAI02870
      I1=N2(K2)                                                         MAI02880
      WRITE(6,2000) TYPE,NAME(I1),(P(J),J=1,JMAX)                       MAI02890
      DO   50 K3=1,IMAX                                                 MAI02900
      A1=IDINT(A(K3))                                                   MAI02910
   50 WRITE(6,2010 )A1,(CFORCM(K3,J,I1),J=1,JMAX)                       MAI02920
   60 CONTINUE                                                          MAI02930
C                                                                       *MAI02940
CC         WRITE THE VECTORS (N1 QUEUE)                                 *MAI02950
C                                                                       *MAI02960
      M1=1                                                              MAI02970
      M2=10                                                             MAI02980
      DO   80 K4=1,3                                                    MAI02990
      IF(N1MAX.LE.10)M2=N1MAX                                           MAI03000
      WRITE(6,2020) TYPE,(NAME(N1(I)),I=M1,M2)                          MAI03010
      DO 65 K6=1,IMAX                                                   MAI03020
      A2=IDINT(A(K6))                                                   MAI03030
   65 WRITE(6,2010)A2,(CFORCM(K6,1,N1(I)),I=M1,M2)                      MAI03040
      M1=M1+10                                                          MAI03050
      IF(M2.EQ.N1MAX)GO TO 90                                           MAI03060
      M2=M2+10                                                          MAI03070
      IF(N1MAX-M2)  70,  80,  80                                        MAI03080
   70 M2=N1MAX                                                          MAI03090
   80 CONTINUE                                                          MAI03100
C                                                                       *MAI03110
CC         LIST THE NULL SETS (NO QUEUE)                                *MAI03120
C                                                                       *MAI03130
   90 WRITE(6,2030)                                                     MAI03140
      M1=1                                                              MAI03150
      M2=7                                                              MAI03160
      DO 110 K7=1,4                                                     MAI03170
      IF(NOMAX.LE.7 )M2=NOMAX                                           MAI03180
      WRITE(6,2040)(NAME(NO(I)),I=M1,M2)                                MAI03190
      M1=M1+7                                                           MAI03200
      IF(M2.EQ.NOMAX)GO TO 120                                          MAI03210
      M2=M2+7                                                           MAI03220
      IF(NOMAX-M2)  100,110,110                                         MAI03230
  100 M2=NOMAX                                                          MAI03240
  110 CONTINUE                                                          MAI03250
  120 PI=3.14159265358979                                               MAI03260
C                                                                       *MAI03270
CC         CONVERT A/B VECTORS TO RADIANS                               *MAI03280
C                                                                       *MAI03290
      DO 130 I=1,IMAX                                                   MAI03300
  130 A(I)=A(I)*PI/180.0                                                MAI03310
      DO  140 J=1,JMAX                                                  MAI03320
  140 B(J)=B(J)*PI/180.0                                                MAI03330
```

```
C                                                                       * MAI03340
CC          READ DATA SET #44                                           * MAI03350
C                                                                       * MAI03360
      READ(5,1002)$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ                    MAI03370
C                                                                       * MAI03380
CCC         NOW SET NSFLAG TO 1 IN ORDER TO ENSURE THAT DATA SET #45     * MAI03390
CCC         IS READ ON THE FIRST PASS                                   * MAI03400
C                                                                       * MAI03410
      NSFLAG=1                                                             MAI03420
C                                                                       * MAI03430
CCC         DO LOOP WHICH PROVIDES OPPORTUNITY FOR THE USER TO INITIATE * MAI03440
CCC         UP TO 1000 ITERATIVE SOLUTIONS                              * MAI03450
C                                                                       * MAI03460
      DO  160 K10=1,1000                                                   MAI03470
      IF((NSFLAG.EQ.2).OR.(NSFLAG.EQ.3))GO TO 155                          MAI03480
C                                                                       * MAI03500
CCC         IF NSFLAG = 2 OR 3,  DATA SETS #46 AND #47 WILL REMAIN      * MAI03510
CCC         CONSTANT AND ONLY DATA SET #48 WILL BE READ                 * MAI03520
C                                                                       * MAI03530
      READ(5,1003,END=170) EPS,DELT,LIMIT,SCF,NPRINT,NSFLAG                MAI03490
      READ(5,1002)R(14),R(15),R(16),AILDEF,RUDDEF,ELEDEF                   MAI03540
C                                                                       * MAI03550
CCC         SAVE THE INITIAL ORIENTATION RATES IF NSFLAG = 3            * MAI03560
C                                                                       * MAI03570
      STHDOT=R(14)                                                         MAI03580
      SPSDOT=R(15)                                                         MAI03590
      SPHDOT=R(16)                                                         MAI03600
  155 READ(5,1004,END= 170)(R(I),I=1,6),R(13),ICCODE                       MAI03610
C                                                                       * MAI03620
CCC         RESET THE INITIAL ORIENTATION RATES                         * MAI03630
C                                                                       * MAI03640
      R(14)=STHDOT                                                         MAI03650
      R(15)=SPSDOT                                                         MAI03660
      R(16)=SPHDOT                                                         MAI03670
C                                                                       * MAI03680
CC          WRITE AIRCRAFT CONSTANTS                                    * MAI03690
C                                                                       * MAI03700
      WRITE(6,2050)TYPE                                                    MAI03710
      WRITE(6,2060)$MASS,CBAR,AREA,$IX,$IY,$IZ,GRAV,RHO                    MAI03720
      WRITE(6,2070)AILDEF,RUDDEF,ELEDEF                                    MAI03730
      IF(NSFLAG.EQ.4) GO TO 156                                            MAI03740
C                                                                       * MAI03750
CC          WRITE PROGRAM CONSTANTS (INTEGRATION SOLUTION METHOD ONLY)  * MAI03760
C                                                                       * MAI03770
      WRITE(6,2080)EPS,DELT,LIMIT                                          MAI03780
      WRITE(6,2090)SCF                                                     MAI03790
C                                                                       * MAI03800
C           WRITE SOLUTION METHOD                                       * MAI03810
```

```
C                                                                       * MAI03820
C                                                                       *
      IF((NSFLAG.EQ.0).OR.(NSFLAG.EQ.2))  WRITE(6,2100)                   MAI03830
      IF((NSFLAG.EQ.1).OR.(NSFLAG.EQ.3)) WRITE(6,2110)                    MAI03840
  156 IF(NSFLAG.EQ.4) WRITE(6,2115)                                       MAI03850
C                                                                       * MAI03860
CCC       CONVERT FROM INPUT UNITS/VARIABLES TO EQUATION COMPATIBLE      * MAI03870
CCC       UNITS/VARIABLES                                               * MAI03880
C                                                                       * MAI03890
      CALL EULER(1)                                                     MAI03900
C                                                                       * MAI03910
CCC       COMPUTE ALPHA AND BETA FROM INITIAL CONDITIONS                * MAI03920
C                                                                       * MAI03930
      CALL EQNS(0,& 160)                                                MAI03940
C                                                                       * MAI03950
CCC       CONVERT EQUATION UNITS/VARIABLES BACK TO OUTPUT VARIABLES/    * MAI03960
CCC       UNITS                                                         * MAI03970
C                                                                       * MAI03980
      CALL EULER(0)                                                     MAI03990
C                                                                       * MAI04000
CCCC      WRITE INITIAL CONDITIONS AND COMPUTED ANGLE OF ATTACK AND     * MAI04010
CCCC      SIDE SLIP ANGLE                                               * MAI04020
C                                                                       * MAI04030
      WRITE(6,2120)ICCODE                                               MAI04040
      WRITE(6,2130)(R(I),I=1,6),ALF,BET                                 MAI04050
      IF((NSFLAG.EQ.0).OR.(NSFLAG.EQ.2)) WRITE(6,2140) R(13)            MAI04060
      IF((NSFLAG.EQ.1).OR.(NSFLAG.EQ.3)) WRITE(6,2150)(R(13),I=13,16)   MAI04070
C                                                                       * MAI04080
CCC       CONVERT UNITS                                                 * MAI04090
C                                                                       * MAI04100
      CALL EULER(1)                                                     MAI04110
C                                                                       * MAI04120
CCC       CHECK ALPHA AND BETA TO SEE IF THEY EXCEED LIMIT OF DATA      * MAI04130
CCC       DEFINITION. IF SO, A MESSAGE WILL BE WRITTEN BY ANGLE AND     * MAI04140
CCC       THE PROGRAM WILL ATTEMPT TO READ THE NEXT DATA SET OF INI-    * MAI04150
CCC       TIAL CONDITIONS DEPENDING ON THE VALUE OF NSFLAG              * MAI04160
C                                                                       * MAI04170
      CALL ANGLE(& 160)                                                 MAI04180
C                                                                       * MAI04190
CCC       NOW FIND A SOLUTION                                           * MAI04200
C                                                                       * MAI04210
      IF(NSFLAG.LE.3) CALL SOLVIT                                       MAI04220
      IF(NSFLAG.EQ.4) CALL GAWAIN                                       MAI04230
C                                                                       * MAI04240
CCC       FAIL SAFE STATEMENT                                           * MAI04250
C                                                                       * MAI04260
      IF(NSFLAG.GE.5) WRITE(6,2160)                                     MAI04270
C                                                                       * MAI04280
```

```
C          GO BACK AND READ THE NEXT INITIAL CONDITION DATA SET                 * MAI04290
C                                                                               * MAI04300
  160 CONTINUE                                                                    MAI04310
  170 STOP                                                                        MAI04320
 1000 FORMAT(A4,2I3,7F10.2/(8F10.2))                                              MAI04330
 1001 FORMAT(I10,10X,A4,10X,I6)                                                   MAI04340
 1002 FORMAT(8F10.6)                                                              MAI04350
 1003 FORMAT(2F10.6,I10,6F5.2,2I10)                                               MAI04360
 1004 FORMAT(7F10.6,I10)                                                          MAI04370
 2000 FORMAT(/////'0',T58,A4,/'0',T52,A4,'(ALPHA,BETA)'//                         MAI04380
     1' ','BETA: ', 9(3X,I3,5X)/  '0','ALPHA' )                                  MAI04390
 2010 FORMAT(' ',I3,10(1X,F10.6))                                                 MAI04400
 2020 FORMAT('0',T51,A4,' F(ALPHA ONLY)'//                                        MAI04410
     1' ','ALPHA',3X,10(A4,7X))                                                   MAI04420
 2030 FORMAT(////'0',T32,'THE BELOW LISTED COEFFICIENTS HAVE BEEN EQUATEMAI04430
     1D TO ZERO',///)                                                            MAI04440
 2040 FORMAT('0',T34,(7(A4,4X)))                                                  MAI04450
 2050 FORMAT('1',T44,'INITIAL INPUT AND COMPUTED DATA'/' ',T49,'FOR THE  MAI04460
     1',A4,' AIRCRAFT'///   ' ',T8,'MASS',T22,'CHORD',T35,'WING AREA',T5MAI04470
     24,'IX',T69,'IY',T84,'IZ',T96,'GRAVITY',T111,'DENSITY'/                      MAI04480
     3' ',T7,'(SLUGS)',T21,'(FEET)',T35,'(FEET**2)',T50,'                (SMAI04490
     4LUG*FEET**2)            ',T94,'(FT/SEC**2)',T107,'(SLUGS/FT**3)')       MAI04500
 2060 FORMAT('0',8(E13.4,2X)///)                                                  MAI04510
 2070 FORMAT('0',T50,'CONTROL DEFLECTIONS'/  ' ',T55,'(DEGREES)'///' ',  MAI04520
     1T36,'AILERON',T57,'RUDDER',T76,'ELEVATOR'// ' ',T37,F6.2,T57,F6.2,MAI04530
     2T77,F6.2/////)                                                             MAI04540
 2080 FORMAT('0',T50,'SOLUTION PARAMETERS',///                                    MAI04550
     1' ',T37,'EPSILON',T58,'DELTA T',T72,'ITERATION LIMIT',/                     MAI04560
     2'0',T37,F7.4,T58,F7.4,T78,I4/////)                                         MAI04570
 2090 FORMAT('0',T46,'SOLUTION CONVERGENCE FACTORS',///                           MAI04580
     1' ',T3,'ALTITUDE RATE',T27,'RADIUS',T46,'SPINRATE',T67,'THETA',    MAI04590
     2T88,'PSI',T108,'PHI'/                                                      MAI04600
     3'0',T8,6(F4.2,16X)/////)                                                   MAI04610
 2100 FORMAT('0',T42,'SOLUTION METHOD:  MODIFIED EQUATIONS'/////)                 MAI04620
 2110 FORMAT('0',T44,'SOLUTION METHOD:  FULL EQUATIONS'/////)                     MAI04630
 2115 FORMAT('0',T47,'SOLUTION METHOD:  GRADIENT'/////)                           MAI04640
 2120 FORMAT('0',T46,'INITIAL CONDITIONS (#',I6,')'///                            MAI04650
     2' ',T2,'SINK RATE',T16,'SPIN RADIUS',T32,'SPIN RATE',T49,'THETA',  MAI04660
     3T65,'PSI',T80,'PHI',T94,'ALPHA',T109,'BETA'/                               MAI04670
     4' ',T3,'(FT/SEC)',T19,'(FEET)',T32,'(DEG/SEC)',T49,'(DEG)',T64,    MAI04680
     5'(DEG)',T79,'(DEG)',T94,'(DEG)',T109,'(DEG)'//)                            MAI04690
 2130 FORMAT(' ',8(F7.2,8X)//)                                                    MAI04700
 2140 FORMAT('0',T55,'RADIUS RATE',/' ',T56,'(FT/SEC)'/'0',T55,F7.2)              MAI04710
 2150 FORMAT('0',T 9,'RADIUS RATE',T40,'THETA RATE',T70,'PSI RATE',T101,MAI04720
     1'PHI RATE',/' ',T11,'(FT/SEC)',T40,'(RAD/SEC)',T70,'(RAD/SEC)',T10MAI04730
     21,'(RAD/SEC)'/'0',T12,4(F7.2,13X))                                         MAI04740
 2160 FORMAT('0',T53,'NSFLAG IS GREATER THAN FOUR')                               MAI04750
      END                                                                        MAI04760
```

```
C*****************************************************************      GAW00010
C                                                                *      GAW00020
C          SUBROUTINE GAWAIN                                     *      GAW00030
C                                                                *      GAW00040
C          PURPOSE:                                              *      GAW00050
C               SOLVES SYSTEM OF NON LINEAR EQUATIONS USING THE OPTIMUM*  GAW00060
C               GRADIENT METHOD.                                 *      GAW00070
C                                                                *      GAW00080
C          METHOD:                                               *      GAW00090
C               A CRITERION FUNCTION IS COMPUTED AS THE 'SUM' OF THE   *  GAW00100
C               ABSOLUTE VALUES OF THE EQUATION RESIDUALS. A GRADIENT  *  GAW00110
C               OF THE CRITERION FUNCTION IS THEN COMPUTED WITH RESPECT*  GAW00120
C               TO CHANGES IN EACH OF THE INDEPENDENT VARIABLES WHICH  *  GAW00130
C               ARE SUBSEQUENTLY INCREMENTED SO AS PROVIDE A SOLUTION  *  GAW00140
C               PATH WHICH WILL MINIMIZE 'SUM'. THE ITERATIVE PROCEDURE*  GAW00150
C               IS CONTINUED UNTIL 'SUM' IS LESS THAN EPS OR UNTIL THE *  GAW00160
C               SOLUTION PATH EXCEEDS THE DATA DEFINITION RANGE OF     *  GAW00170
C               ALPHA OR BETA.                                   *      GAW00180
C                                                                *      GAW00190
C          USAGE:                                                *      GAW00200
C               CALL GAWAIN                                      *      GAW00210
C                                                                *      GAW00220
C          DESCRIPTION OF PARAMETERS:                            *      GAW00230
C               SUM        CRITERION FUNCTION: SUM OF THE ABSOLUTE    *  GAW00240
C                          VALUES OF THE EQUATION RESIDUALS           *  GAW00250
C               FAC        VALUE EQUAL TO 5*DELX(I); USED TO DETERMINE*  GAW00260
C                          GRADIENT OF SUM                            *  GAW00270
C               PRSX(I):   PARTIAL DERIVATIVE OF SUM WITH RESPECT TO FAC* GAW00280
C               DELX(I):   INCREMENT COMPUTED AS A FUNCTION OF THE GRA- *  GAW00290
C                          DIENT OF SUM; ENSURES A CHANGE IN THAT PARTI-*  GAW00300
C                          CULAR INDEPENDENT VARIABLE WHICH WILL YIELD *  GAW00310
C                          A SUBSEQUENT REDUCTION IN SUM.             *  GAW00320
C               K:         COEFFICIENT USED TO CONTROL THE INCREMENT   *  GAW00330
C                          STEP SIZE                                  *  GAW00340
C               M2:        INTEGER EQUAL TO THE ITERATION NUMBER WHEN *  GAW00350
C                          THE MOST RECENT COMPUTATION OF THE GRADIENT*  GAW00360
C                          OCCURRED                                   *  GAW00370
C               M1:        ITERATION NUMBER                           *  GAW00380
C                                                                *      GAW00390
C          COMMON VARIABLES UTILIZED (DESCRIBED IN MAIN PROGRAM) *      GAW00400
C               R                VEL                             *      GAW00410
C               ALF/BET                                          *      GAW00420
C               EPS                                              *      GAW00430
C                                                                *      GAW00440
C          SUBROUTINES REQUIRED                                  *      GAW00450
C               EQNS                                             *      GAW00460
C               EULER                                            *      GAW00470
C*****************************************************************      GAW00490
```

```
      SUBROUTINE GAWAIN                                                  GAW00500
      IMPLICIT REAL*8(A-H,O-Z,$)                                         GAW00510
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,    GAW00520
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF            GAW00530
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,    GAW00540
     1EPS,SUM ,VEL                                                       GAW00550
      DIMENSION      DELX(6),PRSX(6)                                     GAW00560
      REAL K                                                             GAW00570
C                                                                        * GAW00580
CC         ENSURE THAT THE DERIVATIVES ARE EQUAL TO ZERO: EXTRANEOUS     * GAW00590
CC         VALUES WILL YIELD ERRONEOUS VALUES OF ALPHA AND BETA IN       * GAW00600
CC         SUBROUTINE EQNS.                                              * GAW00610
C                                                                        * GAW00620
      DO 5 I=13,16                                                       GAW00630
    5 R(I)=0.0                                                           GAW00640
      WRITE(6,998)                                                       GAW00650
C                                                                        * GAW00660
CC         INITIALIZE DELX(I)                                            * GAW00670
C                                                                        * GAW00680
      DO 10 I=1,6                                                        GAW00690
   10 DELX(I)=0.01*R(I)                                                  GAW00700
      SUM=1.0                                                            GAW00710
C                                                                        * GAW00720
CC         ITERATION DO LOOP                                             * GAW00730
C                                                                        * GAW00740
      DO 60 M1=1,LIMIT                                                   GAW00750
      SUMOLD=SUMALL                                                      GAW00760
C                                                                        * GAW00770
CC         COMPUTE CRITERION FUNCTION                                    * GAW00780
C                                                                        * GAW00790
      CALL EQNS(M,&70)                                                   GAW00800
      SUMALL=SUM                                                         GAW00910
C                                                                        GAW00810
CC         WRITE ITERATION VALUES                                        GAW00820
C                                                                        GAW00830
      CALLEULER(0)                                                       GAW00840
      WRITE(6,999)M1,(R(I),I=1,6),ALF,BET,(R(I),I=7,12),SUM,VEL          GAW00850
      CALL EULER(1)                                                      GAW00860
C                                                                        * GAW00870
CC         CHECK FOR CONVERGENCE/SOLUTION                                * GAW00880
C                                                                        * GAW00890
      IF(SUM.LE.EPS) GO TO 70                                            GAW00900
C                                                                        * GAW00920
CC         IF PREVIOUS SUM IS GREATER THAN CURRENT SUM THE GRADIENT IS   * GAW00930
CC         STILL NEGATIVE THERFORE DELX(I) WILL NOT BE RECOMPUTED. IF,   * GAW00940
C          HOWEVER THE GRADIENT IS POSITIVE RECOMPUTE DELX(I).            * GAW00950
```

```
      IF(SUMOLD-SUM)20,50,50                                                GAW00970
   20 M2=M1                                                                 GAW00980
C                                                                        *  GAW00990
CC         COMPUTE PARTIAL DERIVATIVES                                   ** GAW01000
C                                                                        *  GAW01010
      DO 30 I=1,6                                                           GAW01020
      FAC=5.0*DELX(I)                                                       GAW01030
C     FAC=DELX(I)                                                           GAW01040
C     FAC=R(I)*0.01                                                         GAW01050
      R(I)=R(I) +FAC                                                        GAW01060
      CALL EQNS(M,&70)                                                      GAW01070
      R(I)=R(I) -FAC                                                        GAW01080
      PRSX(I)=(SUM  -SUMALL)/FAC                                            GAW01090
   30 CONTINUE                                                              GAW01100
C                                                                        *  GAW01110
CC         COMPUTE K                                                     ** GAW01120
C                                                                        *  GAW01130
      SUMSQR=0.0                                                            GAW01140
      DO 40 I=1,6                                                           GAW01150
      SUMSQR=SUMSQR + PRSX(I)**2                                            GAW01160
   40 CONTINUE                                                              GAW01170
      K=-SUMALL/SUMSQR                                                      GAW01180
C                                                                        *  GAW01190
CC         WRITE ITERATION VALUES                                        ** GAW01200
C                                                                        *  GAW01210
   50 CALL EULER(0)                                                         GAW01220
      WRITE(6,1000) M2, DELX,PRSX,K                                         GAW01230
      CALL EULER(1)                                                         GAW01240
C                                                                        *  GAW01250
CC         COMPUTE DELX(I)                                               ** GAW01260
CC         INCREMENT INDEPENDENT VARIABLE, R(I)                          ** GAW01270
C                                                                        *  GAW01280
      DO 60 I=1,6                                                           GAW01290
      DELX(I) =K*PRSX(I)                                                    GAW01300
      R(I)=R(I) +DELX(I)                                                    GAW01310
   60 CONTINUE                                                              GAW01320
   70 RETURN                                                                GAW01330
  998 FORMAT('1','M1',   T5,'ALTITUDE RATE',T26,'RADIUS',T42,'SPIN RATE'GAW01340
     1,T61,'THETA',T79,'PSI',T96,'PHI',T112,'ALPHA',T129,'BETA'/          GAW01350
     2' ',T3,'M2',    T8 ,'(FT/SEC)',T26,'(FEET)',T42,'(DEG/SEC)',T61,   GAW01360
     3'(DEG)',T78,'(DEG)',T95,'(DEG)',T112,'(DEG)',T129,'(DEG)'/          GAW01370
     4' ',T18,6('RESID',12X),T121,'SUM'/                                  GAW01380
     5' ',T18,5('DELX',13X), 'DELX ',T119,'VELOCITY'/                     GAW01390
     6' ',T18,5('PRSX',13X),'PRSX',T122,'K')                              GAW01395
  999 FORMAT('0',I4,2X,F7.2,7(10X,F7.2)/                                  GAW01400
     1' ',14X,E9.3,5(8X,E9.3)/' ',T118,E9.3)                              GAW01410
 1000 FORMAT('+',4X,I2,8X,E9.3,5(8X,E9.3)/' ',14X,E9.3,6(8X,E9.3))        GAW01430
      END                                                                  GAW01440
```

```
C*****************************************************************      SOL00010
C                                                                *      SOL00020
C          SUBROUTINE SOLVIT                                     *      SOL00030
C                                                                *      SOL00040
C          PURPOSE                                               *      SOL00050
C               SOLVES EQUATIONS USING INTEGRATION METHOD        *      SOL00060
C                                                                *      SOL00070
C          METHOD                                                *      SOL00080
C               ITERATIVE SOLUTION WHEREBY THE INDEPENDENT VARIABLE      SOL00090
C               ACCELERATIONS ARE COMPUTED FROM THE GIVEN INITIAL *      SOL00100
C               CONDITIONS.  NEW INITIAL CONDITIONS ARE DERIVED BY *     SOL00110
C               INTEGRATING THE ACCELERATIONS AND RATES. A SOLUTION *    SOL00120
C               IS OBTAINED WHEN THE ACCELERATIONS REACH A MINIMUM *     SOL00130
C               VALUE                                            *      SOL00140
C                                                                *      SOL00150
C          APPLICABILITY                                         *      SOL00160
C               FULL AND MODIFIED EQUATIONS ONLY                 *      SOL00170
C                                                                *      SOL00180
C          USAGE                                                 *      SOL00190
C               CALL SOLVIT (INITIAL CONDITIONS MUST HAVE BEEN CON-*     SOL00200
C               VERTED TO EQUATION COMPATIBLE FORM PRIOR TO THE CALL;*   SOL00210
C               (SEE SUBROUTINE EULER)                           *      SOL00220
C                                                                *      SOL00230
C          DESCRIPTION OF PARAMETERS                             *      SOL00240
C               SUMMIN:  MINIMUM SUM COMPUTED BY EQNS            *      SOL00250
C               M1:      ITERATION DO LOOP INDEX                 *      SOL00260
C               MS:      INTEGER EQUAL TO THE VALUE OF M1 WHEN SUMMIN *  SOL00270
C                        IS COMPUTED                             *      SOL00280
C               SAVE:    VECTOR OF VALUES OF THE INDEPENDENT/DEPENDENT *SOL00290
C                        VARIABLES CORRESPONDING TO ITERATION #MS. THIS*  SOL00300
C                        REPRESENTS THE BEST SOLUTION CONVERGENCE.*      SOL00310
C               ZZ:      MATRIX USED TO STORE A SUMMARY OF THE ITERA-*   SOL00320
C                        TIVE SOLUTION. APPROPRIATE VARIABLES ARE *      SOL00330
C                        ASSIGNED TO ZZ EVERY 25 ITERATIONS       *      SOL00340
C                                                                *      SOL00350
C          COMMON VARIABLES UTILIZED (DESCRIBED IN MAIN PROGRAM) *      SOL00360
C               R              SUM              ICCODE           *      SOL00370
C               ALF/BET        DELT             NFLAG            *      SOL00380
C               NPRINT         SCF              LIMIT            *      SOL00390
C                                                                *      SOL00400
C          SUBROUTINES REQUIRED                                  *      SOL00410
C               ANGLE                                            *      SOL00420
C               EULER                                            *      SOL00430
C               EQNS                                             *      SOL00440
C                                                                *      SOL00450
C*****************************************************************      SOL00460
```

```
      SUBROUTINE SOLVIT                                                 SOL00470
      IMPLICIT REAL *8 (A-H,O-Z,$)                                      SOL00480
      COMMON NFLAG,NPRINT,ICCODE                                        SOL00490
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,   SOL00500
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF           SOL00510
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,   SOL00520
     1EPS,SUM ,VEL                                                      SOL00530
      DIMENSION ZZ(81,10),SAVE(10)                                      SOL00540
      M2=1                                                              SOL00550
      SUM=0.0                                                           SOL00560
      SUMMIN=1000.0                                                     SOL00570
      II=1                                                              SOL00580
      MM=0                                                              SOL00590
C                                                                       *SOL00600
CC          INITIALIZE ACCELERATIONS                                    **SOL00610
C                                                                       *SOL00620
      DO 5 I=7,12                                                       SOL00630
    5 R(I)=0.0                                                          SOL00640
C                                                                       *SOL00650
CC          STORE INITIAL CONDITIONS IN ZZ                              **SOL00660
C                                                                       *SOL00670
      CALL EULER(0)                                                     SOL00680
      ZZ(II,1)=FLOAT(MM)                                                SOL00690
      DO 10 J=2,7                                                       SOL00700
   10 ZZ(II,J)=R(J-1)                                                   SOL00710
      ZZ(II,8)=ALF                                                      SOL00720
      ZZ(II,9)=BET                                                      SOL00730
      ZZ(II,10)=VEL                                                     SOL00740
      CALL EULER(1)                                                     SOL00750
      MM=25                                                             SOL00760
C                                                                       *SOL00770
CC          IF NPRINT = 1 SUPPRESS ITERATION PRINTOUT                   **SOL00780
C                                                                       *SOL00790
      IF(NPRINT.EQ.1) GO TO 20                                          SOL00800
      WRITE(6,998)                                                      SOL00810
C                                                                       *SOL00820
CC          PRIMARY INTERATION DO LOOP                                  **SOL00830
C                                                                       *SOL00840
   20 DO  80 M1=1,LIMIT                                                 SOL00850
C                                                                       *SOL00860
CC          COMPUTE ACCELERATIONS AND THE APPLICABLE RATES OF THE       SOL00870
CC          INDEPENDENT VARIABLES                                       SOL00880
C                                                                       *SOL00890
      CALL EQNS(M2,&90)                                                 SOL00900
C                                                                       *SOL00910
CC          CHECK FOR SOLUTION                                          **SOL00920
C                                                                       *SOL00930
      IF(SUM.LE.EPS)GO TO 90                                            SOL00940
```

```
C                                                                       * SOL00950
CCC         WRITE THE INDEPENDENT VARIABLES, ASSOCIATED DERIVATIVES, AND* SOL00960
CCC         DEPENDENT VARIABLES ON EVERY ITERATION UNLESS NPRINT = 1    * SOL00970
C                                                                       * SOL00980
      CALL EULER(0)                                                       SOL00990
      IF(NPRINT.EQ.1) GO TO 30                                            SOL01000
      WRITE(6,999) M1,(R(I),I=1,6),ALF,BET,M2,(R(I),I=7,12),SUM,          SOL01010
     1(R(I),I=13,16),VEL                                                  SOL01020
C                                                                       * SOL01030
CCC         STORE VALUES OF THE SPECIFIED VARIABLES EVERY 25TH ITERATION* SOL01040
C                                                                       * SOL01050
   30 IF(M1.NE.MM) GO TO 50                                               SOL01060
      II=II+1                                                             SOL01070
      ZZ(II ,1)=FLOAT(MM)                                                 SOL01080
      DO 40 J=2,7                                                         SOL01090
   40 ZZ(II,J)=R(J-1)                                                     SOL01100
      ZZ(II,8)=ALF                                                        SOL01110
      ZZ(II,9)=BET                                                        SOL01120
      ZZ(II,10)=VEL                                                       SOL01130
      MM=MM+25                                                            SOL01140
   50 IF(SUM.GT.SUMMIN)GO TO 70                                           SOL01150
C                                                                       * SOL01160
CCC         IF COMPUTED SUM IN LESS THAN SUMMIN, REASSIGN SUMMIN/SAVE/MS* SOL01170
C                                                                       * SOL01180
      MS=M1                                                               SOL01190
      DO 60 I=1,6                                                         SOL01200
   60 SAVE(I)=R(I)                                                        SOL01210
      SAVE(7)=R(13)                                                       SOL01220
      SAVE(8)=R(14)                                                       SOL01230
      SAVE(9)=R(15)                                                       SOL01240
      SAVE(10)=R(16)                                                      SOL01250
      SUMMIN=SUM                                                          SOL01260
   70 CALL EULER(1)                                                       SOL01270
C                                                                       * SOL01280
CCC         INTEGRATE DERIVATIVES/COMPUTE NEW INDEPENDENT VARIABLES     * SOL01290
C                                                                       * SOL01300
      R(1)=R(1) + R(7)*DELT*SCF(1)                                        SOL01310
      R(2)=R(2) +(R(8)*DELT**2/2 +R(13)*DELT)*SCF(2)                      SOL01320
      R(3)=R(3) + R(9)*DELT*SCF(3)                                        SOL01330
      R(4)=R(4) +(R(10)*DELT**2/2 +R(14)*DELT)*SCF(4)                     SOL01340
      R(5)=R(5)+(R(11)*DELT**2/2 +R(15)*DELT)*SCF(5)                      SOL01350
      R(6)=R(6) +(R(12)*DELT**2/2+R(16)*DELT)*SCF(6)                      SOL01360
C                                                                       * SOL01370
CCC         DETERMINE WHETHER CURRENT VALUES OF THETA, PSI, AND PHI HAVE* SOL01380
CCC         EXCEEDED 2*PI; IF SO VALUES ARE APPROPRIATELY REDUCED       * SOL01390
C                                                                       * SOL01400
      CALL ANGLE (&90)                                                    SOL01410
   80 CONTINUE                                                            SOL01420
```

```
C                                                                       * SOL01430
CC          WRITE SOLUTION SUMMARY                                      * SOL01440
C                                                                       * SOL01450
   90 WRITE(6,2000) ICCODE                                                SOL01460
      DO 100 I=1,II                                                       SOL01470
      MO=IDINT(ZZ(I,1))                                                   SOL01480
  100 WRITE(6,2001)MO,(ZZ(I,J),J=2,10)                                    SOL01490
C                                                                       * SOL01500
CC          WRITE BEST CONVERGENCE                                      * SOL01510
C                                                                       * SOL01520
      WRITE(6,2002)                                                       SOL01530
      WRITE(6,2003)MS,(SAVE(J),J=1,6),SUMMIN                              SOL01540
C                                                                       * SOL01550
CC          WRITE MESSAGE IF THE SOLUTION WAS ABNORMALLY TERMINATED     * SOL01560
C                                                                       * SOL01570
      IF(NFLAG.EQ.2) WRITE(6,4000)ALF                                     SOL01580
      IF(NFLAG.EQ.3) WRITE(6,4001)   ALF                                  SOL01590
      IF(NFLAG.EQ.4) WRITE(6,4002)  BET                                   SOL01600
      IF(NFLAG.EQ.5) WRITE(6,4003)  BET                                   SOL01610
      IF(NFLAG.EQ.1) WRITE(6,4004)                                        SOL01620
      RETURN                                                              SOL01630
  998 FORMAT('1','M1',   T5,'ALTITUDE RATE',T26,'RADIUS',T42,'SPIN RATE'SOL01640
     1,T61,'THETA',T79,'PSI',T96,'PHI',T112,'ALPHA',T129,'BETA'/           SOL01650
     2' ',T3,'M2',    T8 ,'(FT/SEC)',T26,'(FEET)',T42,'(DEG/SEC)',T61,   SOL01660
     3'(DEG)',T78,'(DEG)',T95,'(DEG)',T112,'(DEG)',T129,'(DEG)'/         SOL01670
     4' ',T18,6('ACCEL',12X),T121,'SUM'/                                 SOL01680
     5' ',T35,'RATE',T69,2('RATE',13X),'RATE',T119,'VELOCITY')           SOL01690
  999 FORMAT('0',I4,2X,F7.2,7(10X,F7.2)/                                  SOL01700
     1' ',4X,I2,8X,E9.3,6(8X,E9.3)/                                       SOL01710
     2' ',31X,E9.3,25X,3(E9.3,8X),E9.3)                                   SOL01720
 2000 FORMAT('1',T53,'SOLUTION SUMMARY FOR IC #',I6,///                   SOL01730
     1'0',T2,'ITERATION',T13,'ALTITUDE RATE',T30,'RADIUS',T41,'SPINRATE'SOL01740
     2,T56,'THETA',T70,'PSI',T83,'PHI',T95 ,'ALPHA',T108,'BETA',T119,     SOL01750
     1'VELOCITY'/                                                         SOL01760
     3' ',T4,'NUMBER',T16,'(FT/SEC)',T30,'(FEET)',T41,'(DEG/SEC)',T56,   SOL01770
     4'(DEG)',T69,'(DEG)',T82,'(DEG)',T95 ,'(DEG)',T108,'(DEG)',T120,    SOL01780
     1'FT/SEC')                                                           SOL01790
 2001 FORMAT('0',T5,I4,T14,8(F8.2,5X),F8.2)                               SOL01800
 2002 FORMAT(' ',/////)                                                   SOL01810
 2003 FORMAT('0',T55,'SOLUTION CONVERGENCE',/'0',T33,'MINIMUM SUM OF THESOL01820
     1 ABSOLUTE VALUES OF THE INDEPENDENT VARIABLES '///'0',T2,'ITERATIOSOL01830
     2N',T15,'ALTITUDE RATE',T34,'RADIUS',T47,'SPIN RATE',T64,'THETA',    SOL01840
     3T80,'PSI',T95,'PHI',T110,'SUM'/' ',T4,'NUMBER',T18,'(FT/SEC)',T34,SOL01850
     4'(FEET)',T47,'(DEG/SEC)',T64,'(DEG)',T79,'(DEG)',T94,'(DEG)'/      SOL01860
     5'0',T6,I3,T16,7(F8.2,7X))                                          SOL01870
 3000 FORMAT(7E10.4,I10)                                                  SOL01880
 4000 FORMAT('0',T35,'SOLUTION TERMINATED WHEN ALPHA EXCEEDED UPPER LIMISOL01890
     1T (',F5.1,' DEGREES)')                                              SOL01900
```

```
4001 FORMAT('0',T35,'SOLUTION TERMINATED WHEN ALPHA EXCEEDED LOWER LIMISOL01910
     1T (',F5.1,' DEGREES)')                                          SOL01920
4002 FORMAT('0',T35,'SOLUTION TERMINATED WHEN BETA EXCEEDED UPPER LIMITSOL01930
     1 (',F5.1,' DEGREES)')                                           SOL01940
4003 FORMAT('0',T35,'SOLUTION TERMINATED WHEN BETA EXCEEDED LOWER LIMITSOL01950
     1 (',F5.1,' DEGREES)')                                           SOL01960
4004 FORMAT('0',T32,'SOLUTION TERMINATED DUE TO OVERFLOW, UNDERFLOW, ORSOL01970
     1 DIVISION BY ZERO')                                             SOL01980
      END                                                             SOL01990
```

```
C*****************************************************************  EQN00010
C                                                                *  EQN00020
C       SUBROUTINE EQNS                                          *  EQN00030
C                                                                *  EQN00040
C       PURPOSE                                                  *  EQN00050
C            COMPUTES RESIDUALS/ACCELERATIONS OF APPROPRIATE EQUA-*  EQN00060
C            TIONS AS DETERMINED BY NSFLAG                       *  EQN00070
C                                                                *  EQN00080
C       USAGE                                                    *  EQN00090
C            CALL EQNS(M,&____)                                  *  EQN00100
C                                                                *  EQN00110
C       DESCRIPTION OF PARAMETERS                                *  EQN00120
C            C:      VECTOR OF INTERPOLATED VALUES OF STABILITY  *  EQN00130
C                    DERIVATIVES                                 *  EQN00140
C            M:      CALLING ARGUMENT WHICH IF EQUAL TO 0 WILL   *  EQN00150
C                    INITIATE A RETURN TO THE CALLING PROGRAM AFTER*EQN00160
C                    THE COMPUTATION OF ALPHA/BETA. ALSO USED AS *  EQN00170
C                    COUPLED EQUATIONS DO LOOP INDEX             *  EQN00180
C            &___:   CORRESPONDS TO THE STATEMENT NUMBER IN THE CALL-*EQN00190
C                    ING PROGRAM TO WHICH THE SUBROUTINE WILL RETURN *EQN00200
C                    IN THE EVENT OF A RETURN 1                  *  EQN00210
C            NFLAG   SET EQUAL TO 1 BY SUBROUTINE FIX WHENEVER THE*  EQN00220
C                    FORTRAN EXTENDED ERROR HANDLING FACILITY DETECTS*EQN00230
C                    AN EXPONENT UNDERFLOW, OVERFLOW OR ATTEMPT TO *  EQN00240
C                    DIVIDE BY ZERO. WHEN NFLAG = 1 IS DETECTED, A *  EQN00250
C                    RETURN 1 IS EXECUTED AND THE SOLUTION IS    *  EQN00260
C                    TERMINATED                                  *  EQN00270
C                                                                *  EQN00280
C       COMMON VARIABLES UTILIZED (EXPLAINED IN MAIN PROGRAM)    *  EQN00290
C            $MASS   AREA    AILDEF    BETA      EPS             *  EQN00300
C            RHO     $IX     RUDDEF    R         SUM             *  EQN00310
C            CBAR    $IY     ELEDEF    SCF       VEL             *  EQN00320
C            GRAV    $IZ     ALPHA     NSFLAG    NFLAG           *  EQN00330
C                                                                *  EQN00340
C       SUBROUTINES REQUIRED                                     *  EQN00350
C            ANGLE                                               *  EQN00360
C            INTER                                               *  EQN00370
C            FIX                                                 *  EQN00380
C                                                                *  EQN00390
C                                                                *  EQN00400
C*****************************************************************  EQN00410

      SUBROUTINE EQNS(M,*)                                          EQN00420
      IMPLICIT REAL *8 (A-H,O-Z,$)                                  EQN00430
      EXTERNAL FIX                                                  EQN00440
      COMMON NFLAG                                                  EQN00450
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,  EQN00460
```

```
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF                    EQN00470
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,            EQN00480
     1EPS,SUM ,VEL                                                               EQN00490
      DIMENSION  C(42)                                                           EQN00500
      NFLAG =0                                                                   EQN00510
C                                                                          *     EQN00520
CCC       COMPUTE THE DIRECTION COSINES                                    *     EQN00530
C                                                                          *     EQN00540
      A11=DCOS(R(6))*DCOS(R(5))-DSIN(R(6))*DSIN(R(5))*DCOS(R(4))                 EQN00550
      A12=DCOS(R(6))*DSIN(R(5))+DSIN(R(6))*DCOS(R(5))*DCOS(R(4))                 EQN00560
      A13=DSIN(R(4))*DSIN(R(6))                                                  EQN00570
      A21=-DSIN(R(6))*DCOS(R(5))-DCOS(R(6))*DSIN(R(5))*DCOS(R(4))                EQN00580
      A22=-DSIN(R(6))*DSIN(R(5))+DCOS(R(6))*DCOS(R(5))*DCOS(R(4))                EQN00590
      A23=DSIN(R(4))*DCOS(R(6))                                                  EQN00600
      A31=DSIN(R(4))*DSIN(R(5))                                                  EQN00610
      A32=-DSIN(R(4))*DCOS(R(5))                                                 EQN00620
      A33=DCOS(R(4))                                                             EQN00630
C                                                                          *     EQN00640
CCC       COMPUTE ANGLE OF ATTACK AND SIDESLIP ANGLE                       *     EQN00650
C                                                                          *     EQN00660
      ALPHA= DATAN((R(3)*R(2)*A32 + R(1)*A33 + R(13)*A31)                        EQN00670
     1/(R(3)*R(2)*A12 + R(1)*A13 + R(13)*A11))                                   EQN00680
      BETA= DATAN((R(3)*R(2)*A22 + R(1)*A23 +R(13)*A21)                          EQN00690
     1/(R(3)*R(2)*A12 + R(1)*A13 + R(13)*A11))                                   EQN00700
C                                                                          *     EQN00710
CCC       RETURN IF CALL IS FROM MAIN                                      *     EQN00720
C                                                                          *     EQN00730
      IF(M.EQ.0)RETURN                                                           EQN00740
C                                                                          *     EQN00750
CCC       CHECK ALPHA/BETA LIMITS                                          *     EQN00760
C                                                                          *     EQN00770
      CALL ANGLE(&90)                                                            EQN00780
C                                                                          *     EQN00790
CCC       INTERPOLATE STABILITY DERIVATIVES                                *     EQN00800
C                                                                          *     EQN00810
      DO 10 NA=1,42                                                              EQN00820
      CALL INTER(NA,C(NA))                                                       EQN00830
   10 CONTINUE                                                                   EQN00840
C                                                                          *     EQN00850
CCC       COMPUTE FORCE AND MOMENT COEFFICIENTS                            *     EQN00860
C                                                                          *     EQN00870
      VEL=DSQRT(R(1)**2 + (R(3)*R(2))**2 + R(13)**2)                             EQN00880
      FAC13=((R(15)+R(3))*A13+R(14)*DCOS(R(6)))*CBAR/2.0/VEL                     EQN00890
      FAC23=((R(15)+R(3))*A23-R(14)*DSIN(R(6)))*CBAR/2.0/VEL                     EQN00900
      FAC33=((R(15)+R(3))*A33 +R(16))*CBAR/2.0/VEL                               EQN00910
      CFX=C(1) +C(25)*FAC13 + C(26)*FAC23 + C(23)*FAC33                          EQN00920
     1+C(27)*AILDEF + C(24)*RUDDEF + C( 7)*ELEDEF                                EQN00930
      CFY=C(2) +C(8 )*FAC13 + C(28)*FAC23 + C(9)*FAC33                           EQN00940
```

```
     1+C(10)*AILDEF + C(11)*RUDDEF + C(29)*ELEDEF                                    EQN00950
      CFZ=C(3) +C(30)*FAC13 + C(31)*FAC23 + C(32)*FAC33                              EQN00960
     1+C(33)*AILDEF + C(34)*RUDDEF + C(12)*ELEDEF                                    EQN00970
      CMX=C(4) + C(13)*FAC13+ C(35)*FAC23 + C(14)*FAC33                              EQN00980
     1+C(15)*AILDEF + C(16)*RUDDEF + C(36)*ELEDEF                                    EQN00990
      CMY=C(5) +C(37)*FAC13 + C(17)*FAC23 + C(38)*FAC33                              EQN01000
     1+C(39)*AILDEF + C(40)*RUDDEF + C(18)*ELEDEF                                    EQN01010
      CMZ=C(6) +C(19)*FAC13 + C(41)*FAC23 + C(20)*FAC33                              EQN01020
     1+C(21)*AILDEF + C(22)*RUDDEF + C(42)*ELEDEF                                    EQN01030
      FAC1=((R(3)*R(2))**2 + R(1)**2 + R(13)**2)                                     EQN01040
      FAC2=RHO * AREA / 2.0/$MASS                                                    EQN01050
      IF(NSFLAG.NE.4) GO TO 30                                                       EQN01060
C                                                                                    * EQN01070
CC        SHORT EQUATIONS                                                            * EQN01080
CC                                                                                   * EQN01090
      R(7)=-GRAV-(CFX*A13 +CFY*A23 + CFZ*A33)*FAC2*FAC1                              EQN01100
      R(8)=R(3)**2*R(2)-(CFX*A11 + CFY*A21 + CFZ*A31)*FAC2*FAC1                      EQN01110
      R(9)=CFX*A12+CFY*A22+CFZ*A32                                                   EQN01120
      R(10)=2.0*($IX*(DSIN(R(6))**2)+$IY*(DCOS(R(6))**2)-$IZ)*R(3)**2                EQN01130
     1*DSIN(R(4))*DCOS(R(4))+FAC1*RHO*AREA*CBAR                                      EQN01140
     2*(CMX*DCOS(R(6))-CMY*DSIN(R(6)))                                               EQN01150
      R(11)=CMX*A13+CMY*A23+CMZ*A33                                                  EQN01160
      R(12)=2.0*($IX-$IY)*R(3)**2*(DSIN(R(4))**2)*DSIN(R(6))*DCOS(R(6)               EQN01170
     1)+RHO*AREA*CBAR*FAC1*CMZ                                                       EQN01180
CC                                                                                   * EQN01190
CC        COMPUTE SUM OF THE ABSOLUTE VALUES OF THE RESIDUALS                        * EQN01200
CC                                                                                   * EQN01210
      SUM=0.0                                                                        EQN01220
      DO 20 I=7,12                                                                   EQN01230
   20 SUM=SUM+DABS(R(I))                                                             EQN01240
      RETURN                                                                         EQN01250
C                                                                                    * EQN01260
CCC       VERTICAL AND RADIAL EQUATIONS (SAME FOR FULL AND MODIFIED                  * EQN01270
CCC       FORM                                                                       * EQN01280
CCC                                                                                  * EQN01290
   30 R(7)=-GRAV-(CFX*A13 +CFY*A23 +CFZ*A33)*FAC2*FAC1                               EQN01300
      IF(SCF(1).EQ.0.0) R(7)=0.0                                                     EQN01310
      R(8)=R(3)**2*R(2)-(CFX*A11 + CFY*A21 + CFZ*A31)*FAC2*FAC1                      EQN01320
      IF(SCF(2).EQ.0.0)R(8)=0.0                                                      EQN01330
CC                                                                                   * EQN01340
CC        COMPUTE NEW RADIUS RATE                                                    * EQN01350
CC                                                                                   * EQN01360
      R(13)=(R(13) +R(8)*DELT)*SCF(2)                                                EQN01370
      FAC1=((R(3)*R(2))**2 + R(1)**2 + R(13)**2)                                     EQN01380
      SUM1=1.0D+25                                                                   EQN01390
      IF(NFLAG.EQ.1) RETURN 1                                                        EQN01400
C                                                                                    * EQN01410
CC                                                                                   *
```

```
C          DO LOOP TO STABILIZE COUPLED ACCELERATIONS                          * EQN01420
C                                                                              * EQN01430
      DO 60 M=1,20                                                               EQN01440
      IF(NFLAG.EQ.1) RETURN 1                                                    EQN01450
      IF((NSFLAG.EQ.1).OR.(NSFLAG.EQ.3))GO TO 40                                 EQN01460
C                                                                              * EQN01470
CC         MODIFIED EQUATIONS                                                  * EQN01480
C                                                                              * EQN01490
      R(9)=((-$IX*(R(11)*A13 + R(10)*DCOS(R(6)))*A13                             EQN01500
     1-$IY*(R(11)*A23 - R(10)*DSIN(R(6)))*A23                                    EQN01510
     2-$IZ*(R(11)*A33+R(12))*A33) -2.0*$MASS*R(2)*R(3)*R(13)                     EQN01520
     3-(R(2)*(CFX*A12 + CFY*A22 + CFZ*A32)                                       EQN01530
     4- CBAR*(CMX*A13 + CMY*A23 + CMZ*A33))*FAC2*$MASS*FAC1)                     EQN01540
     5/($MASS*R(2)**2 + $IX*A13**2 + $IY*A23**2+$IZ*A33**2)                      EQN01550
      IF(SCF(3).EQ.0.0) R(9)=0.0                                                 EQN01560
      R(10)=(-(R(11) + R(9))*($IX*A13*DCOS(R(6))-$IY*A23*DSIN(R(6)))             EQN01570
     1+ $IX*(R(3)**2)*A13*DCOS(R(4))*DSIN(R(6))                                  EQN01580
     2+$IY*(R(3)**2)*A23*DCOS(R(4))*DCOS (R(6))                                  EQN01590
     3-$IZ*(R(3)**2)*A33*DSIN(R(4))                                              EQN01600
     4+(CMX*DCOS(R(6))-CMY*DSIN(R(6)))                                           EQN01610
     6*FAC2*CBAR*$MASS*FAC1)                                                     EQN01620
     7/($IX*DCOS(R(6))**2+$IY*DSIN(R(6)) **2)                                    EQN01630
      IF(SCF(4).EQ.0.0) R(10)=0.0                                                EQN01640
      R(11)=(-$IX*(R(9)*A13 + R(10)*DCOS(R(6)))*A13                              EQN01650
     1-$IY*(R(9)*A23 - R(10)*DSIN(R(6)))*A23                                     EQN01660
     2-$IZ*(R(9)*A33+R(12))*A33                                                  EQN01670
     3+(CMX*A13 + CMY*A23 + CMZ*A33)                                             EQN01680
     7*FAC2*CBAR*$MASS*FAC1)                                                     EQN01690
     6/($IX*A13**2 + $IY*A23**2 + $IZ*A33**2)                                    EQN01700
      IF(SCF(5).EQ.0.0) R(11)=0.0                                                EQN01710
      R(12)=(($IX-$IY)*R(3)**2*A13*A23-$IZ*(R(11)+R(9))*A33                      EQN01720
     1+CMZ*FAC1*FAC2*$MASS*CBAR)/$IZ                                             EQN01730
      IF(SCF(6).EQ.0.0) R(12)=0.0                                                EQN01740
      IF((NSFLAG.EQ.0).OR.(NSFLAG.EQ.2))GO TO 50                                 EQN01750
C                                                                              * EQN01760
CCC        FULL EQUATIONS                                                      * EQN01770
C                                                                              * EQN01780
   40 R(9)=(-2.0*($MASS*R(3)*R(2)*R(13) + (R(15)+R(3))                         EQN01790
     9*(($IX-$IY)*DCOS(R(6))*A13*R(16)                                         EQN01800
     1+($IX*DSIN(R(6))**2+ $IY*DCOS(R(6))**2-$IZ)*A33*R(14) )                  EQN01810
     2*DSIN(R(4))) - R(11)*($IX*A13**2 +$IY*A23**2 + $IZ*A33**2)               EQN01820
     3-R(10)   *($IX-$IY)*A13*DCOS(R(6))-R(12)*$IZ*A33                         EQN01830
     4+(($IX-$IY)*(DSIN(R(6))**2 -DCOS(R(6))**2)+$IZ)*R(16)*R(14) *             EQN01840
     5DSIN(R(4))-($IX-$IY)*DSIN(R(6))*DCOS(R(6))*DCOS(R(4))*R(14)               EQN01850
     6**2-(R(2)*(CFX*A12 + CFY*A22 + CFZ*A32)                                   EQN01860
     7- CBAR*(CMX*A13 + CMY*A23 + CMZ*A33))*FAC2*$MASS*FAC1)                    EQN01870
     8/($MASS*R(2)**2 + $IX*A13**2 + $IY*A23**2+$IZ*A33**2)                     EQN01880
      IF(SCF(3).EQ.0.0)R(9)=0.0                                                  EQN01890
```

```
      R(10)=((R(15)+ R(3))*($IX*DSIN(R(6))**2+$IY*DCOS(R(6))**2 -$IZ)    EQN01900
     1*(R(16) +(R(15)+R(3))*DCOS(R(4)))*DSIN(R(4))                      EQN01910
     2+2.0*($IX-$IY)*R(16)*R(14) *DSIN(R(6))*DCOS(R(6))                 EQN01920
     3-(R(15)+R(3))*($IX*DCOS(R(6))**2 + $IY*DSIN(R(6))**2)*DSIN(R(4    EQN01930
     4))*R(16)-($IX-$IY)*(R(9)+R(11))*DSIN(R(4))*DSIN(R(6))*DCOS(R(6)   EQN01940
     5)+(CMX*DCOS(R(6))-CMY*DSIN(R(6)))                                 EQN01950
     6*FAC2*CBAR*$MASS*FAC1)                                            EQN01960
     7/($IX*DCOS(R(6))**2+$IY*DSIN(R(6)) **2)                           EQN01970
      IF(SCF(4).EQ.0.0)R(10)=0.0                                        EQN01980
      R(11)=(-2.0*(R(15)+R(3))*(($IX-$IY)*A13*A23*R(16)                 EQN01990
     1+($IX*DSIN(R(6))**2+$IY*DCOS(R(6))**2-$IZ)*DSIN(R(4))*DCOS(R(4    EQN02000
     2))*R(14) )-($IX-$IY)*R(10)*A13*DCOS(R(6))-$IZ*R(12)*A33           EQN02010
     3-R(9)*($IX*A13**2+$IY*A23**2+$IZ*A33**2)                          EQN02020
     4+(($IX-$IY)*(DSIN(R(6))**2-DCOS(R(6))**2)+$IZ)*DSIN(R(4))*R(16)   EQN02030
     5*R(14) -($IX-$IY)*DSIN(R(6))*DCOS(R(6))*DCOS(R(4))*R(14) **2      EQN02040
     3+(CMX*A13 + CMY*A23 + CMZ*A33)                                    EQN02050
     7*FAC2*CBAR*$MASS*FAC1)                                            EQN02060
     8/($IX*A13**2 + $IY*A23**2 + $IZ*A33**2)                           EQN02070
      IF(SCF(5).EQ.0.0)R(11)=0.0                                        EQN02080
      R(12)=(-$IZ*(R(11)+R(9))*A33 + ($IX-$IY)*((R(15)+R(3))*A13        EQN02090
     1+R(14)*DCOS(R(6)))*((R(15)+R(3))*A23-R(14)*DSIN(R(6)))            EQN02100
     2+CMZ*FAC1*FAC2*$MASS*CBAR)/$IZ                                    EQN02110
      IF(SCF(6).EQ.0.0)R(12)=0.0                                        EQN02120
   50 SUM2=SUM1                                                         EQN02130
      SUM1=DABS(R(9))+DABS(R(10))+DABS(R(11))+DABS(R(12))               EQN02140
C                                                                       * EQN02150
C         IF THE SUM OF THE ABSOLUTE VALUES OF THE COUPLED ACCELERA-    * EQN02160
C         TIONS DIFFERS FROM THAT OF THE PREVIOUS ITERATION BY LESS     * EQN02170
C         THAN EPS, JUMP OUT OF DO LOOP                                 * EQN02180
C                                                                       * EQN02190
      IF(DABS(SUM2-SUM1).LE.EPS) GO TO 70                               EQN02200
   60 CONTINUE                                                          EQN02210
C                                                                       * EQN02220
C         COMPUTE SUM OF ALL ACCELERATIONS                              * EQN02230
C                                                                       * EQN02240
   70 SUM=SUM1 +DABS(R(7)) +DABS(R(8))                                  EQN02250
      IF(NFLAG.EQ.1) RETURN 1                                           EQN02260
      IF((NSFLAG.EQ.0).OR.(NSFLAG.EQ.2))GO TO 80                        EQN02270
C                                                                       * EQN02280
C         COMPUTE ORIENTATION ANGLE RATES                               * EQN02290
C                                                                       * EQN02300
      R(14)=(R(14)+R(10)*DELT)*SCF(4)                                   EQN02310
      R(15)=(R(15) +R(11)*DELT)*SCF(5)                                  EQN02320
      R(16)=(R(16)+R(12)*DELT)*SCF(6)                                   EQN02330
   80 RETURN                                                            EQN02340
   90 RETURN 1                                                          EQN02350
      END                                                               EQN02360
```

```
C*****************************************************************************  ANG00010
C                                                                             *  ANG00020
C          SUBROUTINE ANGLE                                                   *  ANG00030
C                                                                             *  ANG00040
C          PURPOSE                                                            *  ANG00050
C              REDUCES MAGNITUDE OF ANGULAR VARIABLES TO LESS THAN 2PI        *  ANG00060
C              INITIATES TERMINATION OF ITERATIVE SOLUTION IF ALPHA OR        *  ANG00070
C              BETA EXCEEDS THEIR RESPECTIVE REGIONS OF DEFINITION            *  ANG00080
C                                                                             *  ANG00090
C          USAGE                                                              *  ANG00100
C              CALL ANGLE(*)                                                  *  ANG00110
C                                                                             *  ANG00120
C          DESCRIPTION OF PARAMETERS                                          *  ANG00130
C              *:  CORRESPONDS TO THE STATEMENT NUMBER IN THE CALLING         *  ANG00140
C                  PROGRAM TO WHICH THE SUBROUTINE WILL RETURN IN THE         *  ANG00150
C                  EVENT OF A RETURN 1                                        *  ANG00160
C              NFLAG:  USED IN THIS SUBROUTINE TO COMMUNICATE WHICH           *  ANG00170
C                      A/B LIMIT HAS BEEN EXCEEDED IN THE EVENT OF AN         *  ANG00180
C                      ABNORMAL TERMINATION                                   *  ANG00190
C                                                                             *  ANG00200
C          COMMON VARIABLES UTILIZED (EXPLAINED IN MAIN PROGRAM)              *  ANG00210
C              R                                                              *  ANG00220
C              A; IMAX; ALPHA                                                 *  ANG00230
C              B; JMAX; BETA                                                  *  ANG00240
C                                                                             *  ANG00250
C*****************************************************************************  ANG00260

      SUBROUTINE ANGLE (*)                                                      ANG00270
      IMPLICIT REAL *8 (A-H,O-Z,$)                                              ANG00280
      COMMON NFLAG                                                              ANG00290
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,           ANG00300
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF                   ANG00310
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,           ANG00320
     1EPS,SUM ,VEL                                                              ANG00330
      DIMENSION Z(5)                                                            ANG00340
      PI=3.14159265358979                                                       ANG00350
      Z(1)=R(4)                                                                 ANG00360
      Z(2)=R(5)                                                                 ANG00370
      Z(3)=R(6)                                                                 ANG00380
      Z(4)=ALPHA                                                                ANG00390
      Z(5)=BETA                                                                 ANG00400
      DO 10 I=1,5                                                               ANG00410
C                                                                               ANG00420
CC        REDUCE TO LESS THAN THE ABSOLUTE VALUE OF 2*PI                        ANG00430
C                                                                               ANG00440
      IF(Z(I))3,1,1                                                             ANG00450
 1    IF(Z(I).LE.2.0 *PI)GO TO 10                                               ANG00460
```

```
      Z(I)=Z(I)-2.0*PI                                                  ANG00470
      IF(Z(I).GT.100.0*PI) Z(I)=Z(I)-98.0*PI                            ANG00480
      GO TO 1                                                           ANG00490
    3 IF(Z(I).GE.-2.0*PI)GO TO 10                                       ANG00500
      Z(I)=Z(I)+2.0*PI                                                  ANG00510
      IF(Z(I).LT.-100.0*PI)Z(I)=Z(I)+98.0*PI                            ANG00520
      GO TO 3                                                           ANG00530
   10 CONTINUE                                                          ANG00540
C                                                                       ANG00550
CC        CHECK WHETHER ALPHA/BETA HAS EXCEEDED UPPER/LOWER BOUNDARIES  ANG00560
CC        OF A/B VECTOR                                                 ANG00570
C                                                                       ANG00580
   11 IF((Z(4).GT.0).AND.(Z(4).GT.A(IMAX)))GO TO 50                     ANG00590
   12 IF((Z(5).GT.0).AND.(Z(5).GT.B(JMAX)))GO TO 51                     ANG00600
   13 IF((Z(4).LE.0).AND.(Z(4).LT.A(1)))GO TO 52                        ANG00610
   14 IF((Z(5).LE.0).AND.(Z(5).LT.B(1)))GO TO 53                        ANG00620
      GO TO 42                                                          ANG00630
C                                                                       ANG00640
CC        WRITE APPLICABLE MESSAGE AND INITIATE TERMINATION OF          ANG00650
CC        ITERATIVE SOLUTION THROUGH RETURN 1 PROCEDURE                 ANG00660
C                                                                       ANG00670
   50 WRITE(6,20)                                                       ANG00680
      NFLAG=2                                                           ANG00690
      ALF=ALPHA*180.0/PI                                                ANG00700
      RETURN 1                                                          ANG00710
   51 WRITE(6,21)                                                       ANG00720
      NFLAG=4                                                           ANG00730
      BET=BETA*180.0/PI                                                 ANG00740
      RETURN 1                                                          ANG00750
   52 WRITE(6,40)                                                       ANG00760
      NFLAG=3                                                           ANG00770
      ALF=ALPHA*180.0/PI                                                ANG00780
      RETURN 1                                                          ANG00790
   53 WRITE(6,41)                                                       ANG00800
      NFLAG=5                                                           ANG00810
      BET=BETA*180.0/PI                                                 ANG00820
      RETURN 1                                                          ANG00830
C                                                                       ANG00840
CC        REASSIGN VALUES                                               ANG00850
C                                                                       ANG00860
   42 R(4)=Z(1)                                                         ANG00870
      R(5)=Z(2)                                                         ANG00880
      R(6)=Z(3)                                                         ANG00890
      ALPHA=Z(4)                                                        ANG00900
      BETA=Z(5)                                                         ANG00910
      RETURN                                                            ANG00920
   20 FORMAT('0',T16,'***** ITERATION TERMINATED: ALPHA HAS EXCEEDED THEANG00930
     1 UPPER LIMIT OF DATA DEFINITION *****')                           ANG00940
```

```
 40    FORMAT('0',T16,'***** ITERATION TERMINATED: ALPHA HAS EXCEEDED THEANG00950
     1 LOWER LIMIT OF DATA DEFINITION *****')                           ANG00960
 21    FORMAT('0',T16,'***** ITERATION TERMINATED: BETA HAS EXCEEDED THE ANG00970
     1UPPER LIMIT OF DATA DEFINITION *****')                            ANG00980
 41    FORMAT('0',T16,'***** ITERATION TERMINATED: BETA HAS EXCEEDED THE ANG00990
     1LOWER LIMIT OF DATA DEFINITION *****')                            ANG01000
       END                                                              ANG01010
C
C
C
C*****************************************************************     FIX00010
C                                                               *     FIX00020
C          SUBROUTINE FIX                                       *     FIX00030
C                                                               *     FIX00040
C          PURPOSE                                              *     FIX00050
C               SETS NFLAG = 1 IN THE EVENT OF A CALL FROM THE EXTENDED* FIX00060
C               ERROR HANDLING FACILITY. A CALL TO ERRSET MUST BE MADE * FIX00070
C               IN THE MAIN PROGRAM IN ORDER TO DETERMINE WHICH ERRORS * FIX00080
C               WILL INITIATE A CALL TO FIX.                         *     FIX00090
C                                                               *     FIX00100
C          DESCRIPTION OF PARAMETERS                            *     FIX00110
C               NFLAG:  INTEGER WHICH IF EQUAL TO ONE, WILL INITITATE *  FIX00120
C                       AN ABNORMAL TERMINATION TO A SOLUTION   *     FIX00130
C               IRETCD: INTEGER WHICH IF EQUAL TO ZERO DIRECTS THE *     FIX00140
C                       EXTENDED ERROR HANDLING FACILITY TO EXECUTE *    FIX00150
C                       STANDARD CORRECTIVE ACTION FOR THE DETECTED *    FIX00160
C                       ERROR                                        FIX00170
C                                                               *     FIX00180
C*****************************************************************     FIX00190

      SUBROUTINE FIX(IRETCD)                                            FIX00200
      COMMON NFLAG                                                      FIX00210
      NFLAG=1,                                                          FIX00220
      IRETCD=0                                                          FIX00230
      RETURN                                                            FIX00240
      END                                                               FIX00250
```

```
C-------------------------------------------------------------------SPL00010
C                                                                   SPL00020
C     SUBROUTINE SPLIN                                              SPL00030
C                                                                   SPL00040
C     PURPOSE                                                       SPL00050
C         PROVIDES INTERPOLATED VALUE USING "CUBIC SPLINE FITTING"  SPL00060
C                                                                   SPL00070
C     USAGE                                                         SPL00080
C         FIRST CALL TO SUBROUTINE:                                 SPL00090
C         CALL SPLIN1(X,Y,M,XINT,YINT)                              SPL00100
C                                                                   SPL00110
C         SUBSEQUENT CALLS:                                         SPL00120
C         CALL SPLINN(X,Y,M,XINT,YINT)                              SPL00130
C                                                                   SPL00140
C     DESCRIPTION OF PARAMETERS                                     SPL00150
C         X:      MONOTONICALLY INCREASING ABSCISSA ARRAY           SPL00160
C         Y:      ONE-FOR-ONE CORRESPONDING ORDINATE ARRAY          SPL00170
C         M:      NUMBER OF X AND Y VALUES SUPPLIED < OR = 300      SPL00180
C         XINT:   VALUE OF ABSCISSA FOR WHICH CORRESPONDING ORDINATE SPL00190
C                 IS TO BE INTERPOLATED (OR EXTRAPOLATED)           SPL00200
C         YINT:   INTERPOLATED (OR EXTRAPOLATED) ORDINATE VALUE     SPL00210
C                                                                   SPL00220
C         X,Y,XINT, AND YINT ARE REAL*8                             SPL00230
C                                                                   SPL00240
C     REMARKS                                                       SPL00250
C         IF SPECIFIED X FALLS OUTSIDE OF RANGE, AN EXTRAPOLATED    SPL00260
C         VALUE WILL BE SUPPLIED                                    SPL00270
C                                                                   SPL00280
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                 SPL00290
C         SUBROUTINE SPLICO IS INCLUDED IN SUBROUTINE SPLIN PACKAGE SPL00300
C                                                                   SPL00310
C     MATHEMATICAL METHOD                                           SPL00320
C         UPON FIRST ENTRY TO SPLIN, A CALL TO SPLICO IS MADE TO    SPL00330
C         DETERMINE THE COEFFICIENTS TO BE USED IN PERFORMING THE   SPL00340
C         INTERPOLATIONS. SEARCH FOR BRACKETING ABSCISSA VALUES IS  SPL00350
C         ALWAYS MADE FROM THE REFERENCE LAST USED IN INTERPOLATING. SPL00360
C                                                                   SPL00370
C     REFERENCE                                                     SPL00380
C         PENNINGTON, RALPH H.,"INTRODUCTORY COMPUTER METHODS AND   SPL00390
C         NUMERICAL ANALYSIS", THE MACMILLAN COMPANY, NEW YORK, 1965 SPL00400
C                                                                   SPL00410
C     SAMPLE PROGRAMS                                               SPL00420
C                                                                   SPL00430
C     1.  GIVEN 10 PAIRS OF ABSCISSA AND ORDINATE VALUES:           SPL00440
C                                                                   SPL00450
C         X=1.,5.,15.,21.,25.,32.,38.,44.,75.,90.                   SPL00460
C         Y=0.,50.,250.,400.,750.,780.,900.,950.,975.,1100.         SPL00470
C                                                                   SPL00480
```

```
C
C       FIND AN ORDINATE VALUE WHEN THE ABSCISSA VALUE IS:                SPL00490
C       20.,60., AND 100.                                                 SPL00500
C                                                                         SPL00510
C       REAL*8 X(10),Y(10),XINT(3)/20.,60.,100./,YINT(3)                  SPL00520
C       CALL SPLIN1(X,Y,10,XINT(1),YINT(1))                               SPL00530
C       DO 20 I=2,3                                                       SPL00540
C    20 CALL SPLINN(X,Y,10,XINT(I),YINT(I))                               SPL00550
C                                                                         SPL00560
C   2.  GIVEN A TWO DIMENSIONAL TABLE WITH N ROWS AND M COLUMNS,          SPL00570
C       N ABSCISSA VALUES, AND M ORDINATE VALUES, DETERMINE A             SPL00580
C       VALUE IN THE TABLE AT A SPECIFIED ABSCISSA AND ORDINATE           SPL00590
C       VALUE.                                                            SPL00600
C                                                                         SPL00610
C       REAL*8 A(10,20),U(10),V(20),YY(20),YI(10),X,Y,FXY                 SPL00620
C       N=10                                                              SPL00630
C       M=20                                                              SPL00640
C       DO 100 NR=1,N                                                     SPL00650
C       DO 101 NC=1,M                                                     SPL00660
C   101 YY(NC)=A(NR,NC)                                                   SPL00670
C   100 CALL SPLIN1(V,YY,M,Y,YI(NR))                                      SPL00680
C       CALL SPLIN1(U,YI,N,X,FXY)                                         SPL00690
C                                                                         SPL00700
C       WHERE A CONTAINS THE TWO DIMENSIONAL TABLE                        SPL00710
C             N IS THE NUMBER OF ROWS IN A                                SPL00720
C             M IS THE NUMBER OF COLUMNS IN A                             SPL00730
C             U CONTAINS THE N ABSCISSA VALUES                            SPL00740
C             V CONTAINS THE M ORDINATE VALUES                            SPL00750
C             YY CONTAINS A SELECTED ROW OF THE TABLE A                   SPL00760
C             YI CONTAINS AN INTERPOLATED VALUE OF A FOR EACH ROW         SPL00770
C                AT THE SELECTED ORDINATE VALUE                           SPL00780
C             X IS THE ABSCISSA FOR WHICH A FUNCTION VALUE                SPL00790
C               IS INTERPOLATED                                           SPL00800
C             Y IS THE ORDINATE FOR WHICH A FUNCTION VALUE IS             SPL00810
C               TO BE RETURNED                                            SPL00820
C             FXY IS A VALUE OF A AT X,Y                                  SPL00830
C                                                                         SPL00840
C                                                                         SPL00850
C                                                                         SPL00860
C                                                                         SPL00870
C-------------------------------------------------------------------------SPL00880
C                                                                         SPL00890

      SUBROUTINE SPLIN1(X,Y,M,XINT,YINT)                                  SPL00900
      IMPLICIT REAL*8 (A-H),REAL*8 (O-Z)                                  SPL00910
      DIMENSION X(M),Y(M),C(4,300)                                        SPL00920
      CALL SPLICO(X,Y,M,C)                                                SPL00930
      K=1                                                                 SPL00940
```

```
      ENTRY SPLINN(X,Y,M,XINT,YINT)                                     SPL00950
    3 IF(XINT-X(1)) 70,1,2                                              SPL00960
   70 K=1                                                               SPL00970
      GO TO 7                                                           SPL00980
    1 YINT=Y(1)                                                         SPL00990
      RETURN                                                            SPL01000
    2 IF(XINT-X(K+1))6,4,5                                              SPL01010
    4 YINT=Y(K+1)                                                       SPL01020
      RETURN                                                            SPL01030
    5 K=K+1                                                             SPL01040
      IF(M-K) 71,71,3                                                   SPL01050
   71 K=M-1                                                             SPL01060
      GO TO 7                                                           SPL01070
    6 IF(XINT-X(K))13,12,11                                             SPL01080
   12 YINT=Y(K)                                                         SPL01090
      RETURN                                                            SPL01100
   13 K=K-1                                                             SPL01110
      GO TO 6                                                           SPL01120
    7 PRINT 101,XINT                                                    SPL01130
  101 FORMAT(8HOXINT = E18.9,32H, OUT OF RANGE FOR INTERPOLATION)       SPL01140
   11 YINT=(X(K+1)-XINT)*(C(1,K)*(X(K+1)-XINT)**2+C(3,K))               SPL01150
      YINT=YINT+(XINT-X(K))*(C(2,K)*(XINT-X(K))**2+C(4,K))              SPL01160
      RETURN                                                            SPL01170
      END                                                               SPL01180

      SUBROUTINE SPLICO(X,Y,M,C)                                        SPL01190
      IMPLICIT REAL*8 (A-H),REAL*8 (O-Z)                                SPL01200
      DIMENSION X(M),Y(M),C(4,300),D(300),P(300),E(300),A(300,3),B(300),SPL01210
     1Z(300)                                                            SPL01220
      MM=M-1                                                            SPL01230
      DO 2 K=1,MM                                                       SPL01240
      D(K)=X(K+1)-X(K)                                                  SPL01250
      P(K)=D(K)/6.                                                      SPL01260
    2 E(K)=(Y(K+1)-Y(K))/D(K)                                           SPL01270
      DO 3 K=2,MM                                                       SPL01280
    3 B(K)=E(K)-E(K-1)                                                  SPL01290
      A(1,2)=-1.-D(1)/D(2)                                              SPL01300
      A(1,3)=D(1)/D(2)                                                  SPL01310
      A(2,3)=P(2)-P(1)*A(1,3)                                           SPL01320
      A(2,2)=2.*(P(1)+P(2))-P(1)*A(1,2)                                 SPL01330
      A(2,3)=A(2,3)/A(2,2)                                              SPL01340
      B(2)=B(2)/A(2,2)                                                  SPL01350
      DO 4 K=3,MM                                                       SPL01360
      A(K,2)=2.*(P(K-1)+P(K))-P(K-1)*A(K-1,3)                           SPL01370
      B(K)=B(K)-P(K-1)*B(K-1)                                           SPL01380
      A(K,3)=P(K)/A(K,2)                                                SPL01390
    4 B(K)=B(K)/A(K,2)                                                  SPL01400
```

```
      Q=D(M-2)/D(M-1)                                                   SPL01410
      A(M,1)=1.+Q+A(M-2,3)                                              SPL01420
      A(M,2)=-Q-A(M,1)*A(M-1,3)                                         SPL01430
      B(M)=B(M-2)-A(M,1)*B(M-1)                                         SPL01440
      Z(M)=B(M)/A(M,2)                                                  SPL01450
      MN=M-2                                                            SPL01460
      DO 6 I=1,MN                                                       SPL01470
      K=M-I                                                             SPL01480
    6 Z(K)=B(K)-A(K,3)*Z(K+1)                                           SPL01490
      Z(1)=-A(1,2)*Z(2)-A(1,3)*Z(3)                                     SPL01500
      DO 7 K=1,MM                                                       SPL01510
      Q=1./(6.*D(K))                                                    SPL01520
      C(1,K)=Z(K)*Q                                                     SPL01530
      C(2,K)=Z(K+1)*Q                                                   SPL01540
      C(3,K)=Y(K)/D(K)-Z(K)*P(K)                                        SPL01550
    7 C(4,K)=Y(K+1)/D(K)-Z(K+1)*P(K)                                    SPL01560
      RETURN                                                            SPL01570
      END                                                               SPL01580
```

```
C*****************************************************************      EUL00010
C          SUBROUTINE EULER                                           *  EUL00020
C                                                                     *  EUL00030
C          PURPOSE                                                    *  EUL00040
C               CONVERTS ANGULAR VARIABLES SO AS TO BE EITHER COMPATI- *  EUL00050
C               BLE  WITH THE PRINTED OUTPUT OR THE APPLICABLE FORM OF *  EUL00060
C               THE EQUATIONS OF MOTION                               *  EUL00070
C                                                                     *  EUL00080
C          USAGE                                                      *  EUL00090
C               CALL EULER(N)                                         *  EUL00100
C                                                                     *  EUL00110
C          DESCRIPTION OF PARAMETERS                                  *  EUL00120
C               N:       INTEGER WHICH DETERMINE WHICH OF TWO POSSIBLE *  EUL00130
C                        DATA CONVERSIONS IS TO BE EMPLOYED            *  EUL00140
C               THETA:   PITCH ORIENTATION ANGLE                       *  EUL00150
C               PSI:     ROLL ORIENTATION ANGLE                        *  EUL00160
C               PHI:     YAW ORIENTATION ANGLE                         *  EUL00170
C                                                                     *  EUL00180
C                        NOTE:  REFER TO REFERENCE (A) OF THE MAIN PRO- *  EUL00190
C                        GRAM FOR ADDITIONAL EXPLAINATION REGARDING THE *  EUL00200
C                        DIFFERENCES BETWEEN THE ORDERED AND REFERENCE  *  EUL00210
C                        EULER ANGLES                                  *  EUL00220
C                        N=0  EQUATION VARIABLES ARE CONVERTED FOR USE  *  EUL00230
C                             AS PRINTED OUTPUT  (ORDERED EULER ANGLES, *  EUL00240
C                             DEGREES)                                  *  EUL00250
C                        N=1  INPUT VARIABLES ARE CONVERTED SO AS TO BE *  EUL00260
C                             COMPATIBLE WITH THE APPLICABLE EQUATIONS  *  EUL00270
C                             (REFERENCE EULER ANGLES, RADIANS)         *  EUL00280
C                                                                     *  EUL00290
C          COMMON VARIABLES UTILIZED                                  *  EUL00300
C               ALPHA/ALF                                             *  EUL00310
C               BETA/BET                                              *  EUL00320
C               R                                                     *  EUL00330
C                                                                     *  EUL00340
C*****************************************************************      EUL00350

      SUBROUTINE EULER(N)                                                EUL00360
      IMPLICIT REAL *8 (A-H,O-Z,$)                                       EUL00370
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,    EUL00380
     1EPS,SUM  ,VEL                                                      EUL00390
      PI=3.14159265358979                                                EUL00400
C                                                                        EUL00410
C          N=0:  CONVERT REFERENCE EULER ANGLES TO ORDERED EULER ANGLES  EUL00420
C                AND THEN CONVERT ANGULAR VARIABLES FROM RADIANS TO      EUL00430
C                DEGREES                                                 EUL00440
C                                                                        EUL00450
C                                                                     *
```

```
C          N=1:  CONVERT ANGULAR VARIABLES FROM DEGREES TO RADIANS AND    EUL00460
CC                THEN CONVERT ORDERED EULER ANGLES TO REFERENCE EULER    EUL00470
CC                ANGLES                                                 EUL00480
C                                                                         EUL00490
      IF(N.EQ.1)GO TO 1                                                   EUL00500
C                                                                         EUL00510
CC         EULER ANGLE CONVERSION: REFERENCE TO ORDERED                   EUL00520
C                                                                         EUL00530
      X=R(4)                                                              EUL00540
      Y=R(5)                                                              EUL00550
      Z=R(6)                                                              EUL00560
      CTERM=DATAN(DTAN(Y/2.0)*DSIN((PI/2.0-X)/2.0)/DSIN((PI/2.0+X)/2.0))  EUL00570
      DTERM=DATAN(DTAN(Y/2.0)*DCOS((PI/2.0-X)/2.0)/DCOS((PI/2.0+X)/2.0))  EUL00580
      PSI=Z+CTERM+DTERM                                                   EUL00590
      THETA=DTERM-CTERM                                                   EUL00600
      PHI=PI/2.0-2.0*DATAN(DSIN(CTERM)/(DSIN(DTERM)*DTAN((PI/2.0-X)/2.0)  EUL00610
     1))                                                                  EUL00620
      R(4)=THETA*180.0/PI                                                 EUL00630
      R(5)=PSI*180.0/PI                                                   EUL00640
      R(6)=PHI*180.0/PI                                                   EUL00650
      R(3)=R(3)*180.0/PI                                                  EUL00660
      ALF=ALPHA*180.0/PI                                                  EUL00670
      BET=BETA*180.0/PI                                                   EUL00680
      GO TO 3                                                             EUL00690
C                                                                         EUL00700
CC         EULER ANGLE CONVERSION: ORDERED TO REFERENCE                   EUL00710
C                                                                         EUL00720
    1 DO 2 I=3,6                                                          EUL00730
    2 R(I)=R(I)*PI/180.0                                                  EUL00740
      X=R(4)                                                              EUL00750
      Y=R(5)                                                              EUL00760
      Z=R(6)                                                              EUL00770
      ATERM=DATAN(DTAN(X/2.0)*DTAN(Z/2.0))                                EUL00780
      BTERM=DATAN(DTAN(X/2.0)/DTAN(Z/2.0))                                EUL00790
      PHI=Y-ATERM-BTERM                                                   EUL00800
      PSI=BTERM-ATERM                                                     EUL00810
      THETA=2.0*DATAN(DSIN(ATERM)/(DSIN(BTERM)*DTAN(Z/2.0)))              EUL00820
      R(4)=THETA                                                          EUL00830
      R(5)=PSI                                                            EUL00840
      R(6)=PHI                                                            EUL00850
    3 RETURN                                                              EUL00860
      END                                                                 EUL00870
```

```
C*****************************************************************************  INT00010
C                                                                             *  INT00020
C          SUBROUTINE INTER                                                   *  INT00030
C                                                                             *  INT00040
C          PURPOSE                                                            *  INT00050
C              INTERPOLATES COEFFICIENT MATRICES OR VECTORS                   *  INT00060
C                                                                             *  INT00070
C          USAGE                                                              *  INT00080
C              CALL INTER(NA,VALUE)                                           *  INT00090
C                                                                             *  INT00100
C          DESCRIPTION OF PARAMETERS                                          *  INT00110
C              NA:      INTEGER CORRESPONDING TO THE COEFFICIENT MATRIX       *  INT00120
C                       OR VECTOR TO BE INTERPOLATED                          *  INT00130
C              VALUE:   INTERPOLATED VALUE OF THE COEFFICIENT MATRIX          *  INT00140
C                                                                             *  INT00150
C          COMMON VARIABLES UTILIZED (DESCRIBED IN MAIN PROGRAM)              *  INT00160
C              CFORCM                                                         *  INT00170
C              A; IMAX; ALPHA                                                 *  INT00180
C              B; JMAX; BETA                                                  *  INT00190
C              KEY(NA)                                                        *  INT00200
C                                                                             *  INT00210
C          UTILITIES/DIMENSIONS                                               *  INT00220
C              AMAT(IMAX,JMAX)                                                *  INT00230
C              YY(JMAX)                                                       *  INT00240
C              YI(IMAX)                                                       *  INT00250
C                                                                             *  INT00260
C          SUBROUTINES REQUIRED                                               *  INT00270
C              SUBROJTINE SPLIN                                               *  INT00280
C                                                                             *  INT00290
C*****************************************************************************  INT00300

      SUBROUTINE INTER(NA,VALUE)                                                 INT00310
      IMPLICIT REAL *8 (A-H,O-Z,$)                                               INT00320
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,            INT00330
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF                    INT00340
      COMMON/WORKB/ALPHA,ALF,BETA,BET,R(16),DELT,SCF(6),LIMIT,NSFLAG,            INT00350
     1EPS,SUM ,VEL                                                               INT00360
      DIMENSION AMAT(22,9),YY(9),YI(22)                                          INT00370
      II=KEY(NA)+1                                                               INT00380
C                                                                                INT00390
C         II=1; UNDEFINED STABILITY DERIVATIVE                                   INT00400
C         II=2; COEFFICIENT VECTOR                                               INT00410
C         II=3; COEFFICIENT MATRIX                                               INT00420
C                                                                                INT00430
      GO TO (1,2,4 ),II                                                          INT00440
C                                                                                INT00450
```

```
C          UNDEFINED STABILITY DERIVATIVES ARE EQUATED TO ZERO                INT00460
C                                                                             INT00470
  1     VALUE=0.0                                                             INT00480
        RETURN                                                                INT00490
C                                                                             INT00500
C           VECTOR INTERPOLATION                                              INT00510
C                                                                             INT00520
  2     DO 3 I=1,IMAX                                                         INT00530
  3     YI(I)=(CFORCM(I,1,NA))                                                INT00540
        CALL SPLIN1(A,YI,IMAX,ALPHA,VALUE)                                    INT00550
        RETURN                                                                INT00560
C                                                                             INT00570
C           MATRIX INTERPOLATION                                              INT00580
C                                                                             INT00590
  4     DO 5  I=1,IMAX                                                        INT00600
        DO 5 J= 1,JMAX                                                        INT00610
  5     AMAT(I,J)=(CFORCM(I,J,NA))                                            INT00620
        DO 7 NR=1,IMAX                                                        INT00630
        DO 6 NC=1,JMAX                                                        INT00640
  6     YY(NC)=AMAT(NR,NC)                                                    INT00650
C                                                                             INT00660
C           USE SPLIN1 TO INTERPOLATE/FILL UTILITY VECTOR (YI) WHICH          INT00670
C           CORRESPONDS TO B=BETA FOR ALL VALUES OF A                         INT00680
C                                                                             INT00690
  7     CALL SPLIN1(B,YY,JMAX,BETA,YI(NR))                                    INT00700
C           ASSIGN VALUES                                                     INT00710
C           NOW INTERPOLATE YI AT A=ALPHA                                     INT00720
C                                                                             INT00730
        CALL SPLIN1(A,YI,IMAX,ALPHA,VALUE)                                    INT00740
        RETURN                                                                INT00750
        END                                                                   INT00760
```

# APPENDIX C

## CRITERION FUNCTION SEARCH PROGRAM

The aircraft spin solution program was modified in order to evaluate the "reasonableness" of various sets of initial conditions. The resulting Criterion Function Search program is designed to compute the criterion value for a given set of initial conditions. The initial conditions are generated by six nested "do loops" within MAIN. The dependent variables $\alpha$ , $\beta$ , V and n are subsequently computed and compared to pre-established constraints; if the constraints are satisfied, the criterion is subsequently computed. If the value is less than $10^4$, the independent variables are printed. An abbreviated flowchart of the program is included as Figure C-1. Pertinent data relating to the initial condition "do loops" and dependent variable constraints is included as Table C-I. The program is basically the same as the spin solution program except that SOLVIT and GAWAIN are not included and that MAIN and EQNS are modified to include the initial condition "do loops" and constraints. The range of the do loops is such that the initial conditions are all within a reasonable approximation to an erect spin.

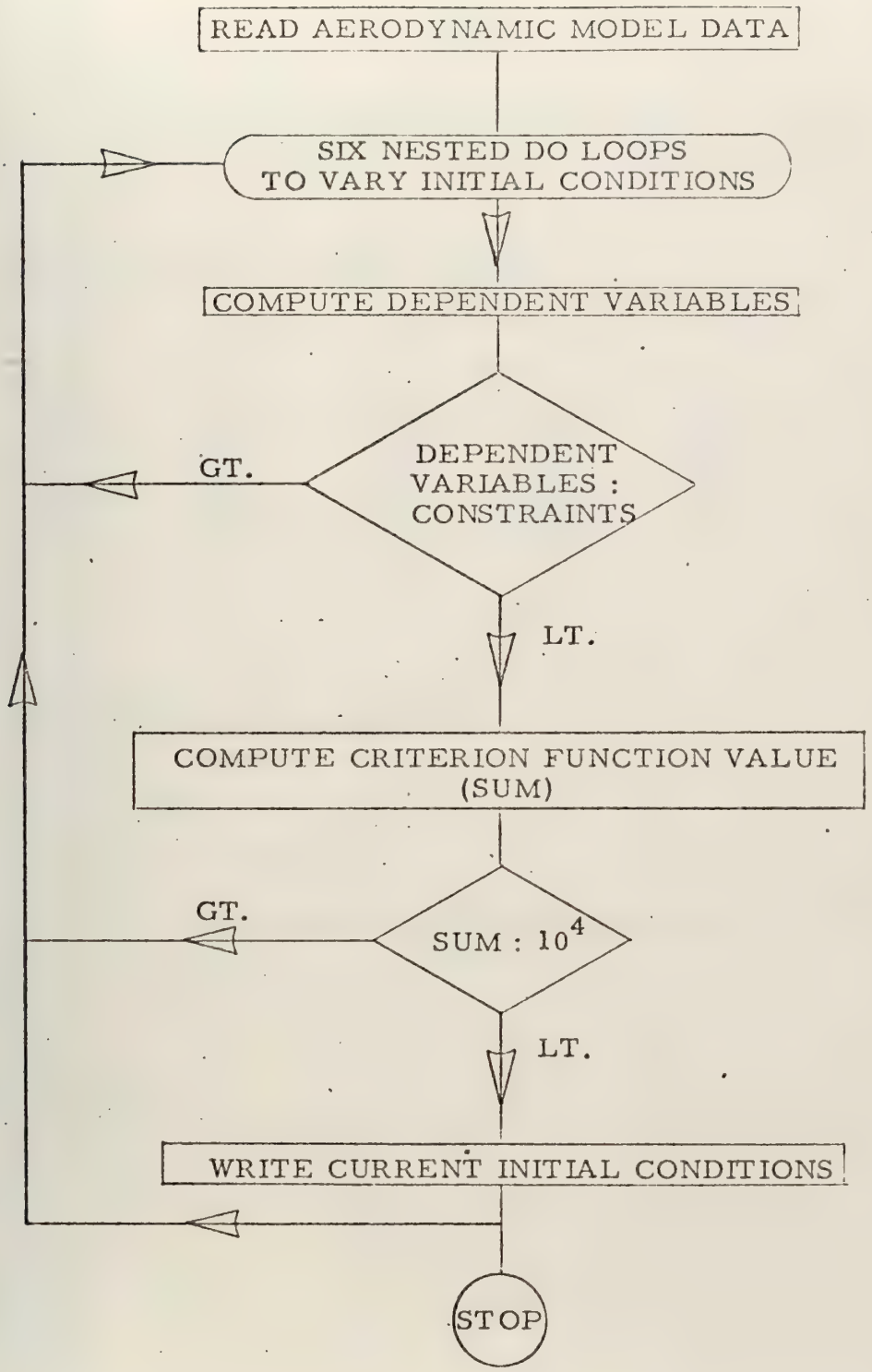



Figure C-1

Abbreviated Flowchart for the Criterion Function Search Computer Program

TABLE C-I

Pertinent Data Regarding the Criterion Function Search Program Independent and Dependent Variables

Initial Condition "Do Loop" Ranges

| $\dot{Z}_o$ | $R$ | $\dot{\gamma}$ | $\Theta_P$ | $\Psi_R$ | $\Phi_Y$ |
|---|---|---|---|---|---|
| -150.0 | 25.0 | 10.0 | -11.0 | 5.0 | - 5.0 |
| -200.0 | 100.0 | 20.0 | -21.0 | 15.0 | -15.0 |
| -250.0 | 175.0 | 30.0 | -31.0 | 25.0 | -25.0 |
| -300.0 | 250.0 | 40.0 | -41.0 | 35.0 | -35.0 |
| -350.0 | 325.0 | 50.0 | -51.0 | 45.0 | -45.0 |
| -400.0 | 400.0 | 60.0 | -61.0 | 55.0 | -55.0 |
| | | | | | -65.0 |
| | | | | | -75.0 |
| | | | | | -85.0 |

Dependent Variable Constraints

$\beta = 0^\circ \pm 40^\circ$

$\alpha = -1.0^\circ$ to $+ 89.0^\circ$

$V_r \leq 589.0$ (corresponds to 300 knots equivalent airspeed)

$n \leq 2.0$ g's

Fixed Constants

$\rho = .001756$ slugs/ft$^3$

| | F-111 | F-4 |
|---|---|---|
| $\delta_a =$ | 10.0 | 21.0 |
| $\delta_r =$ | -15.0 | -30.0 |
| $\delta_e =$ | -25.0 | -30.0 |

```
      IMPLICIT REAL*8(A-H,O-Z,$)                                        MAI00010
      EXTERNAL FIX                                                      MAI00020
      COMMON NFLAG                                                      MAI00030
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,   MAI00040
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF
      COMMON/WORKB/ALPHA,BETA,R(12)
      REAL NAME(42) ,TYPE                                               MAI00070
      INTEGER P(9),A1,A2                                                MAI00080
      DIMENSION N1(42),N3(42),F(12)                                     MAI00090
      CALL ERRSET(207,100,-1,1,FIX,209)
      READ(5,1010) TYPE,IMAX,JMAX,(A(II),II=1,IMAX),(B(J),J=1,JMAX)     MAI00110
1010  FORMAT(A4,2I3,7F10.2/(8F10.2))                                    MAI00120
      N1CNTR=0                                                          MAI00130
      N3CNTR=0                                                          MAI00140
      DO 121 K1=1,42                                                    MAI00150
      READ (5,1009)NA,NAME(NA),KEY(NA)                                  MAI00160
      WRITE(6,1009) NA,NAME(NA),KEY(NA)                                 MAI00170
      IF(KEY(NA).EQ.1)READ(5,1000)(CFORCM(I,1,NA),I=1,IMAX)             MAI00180
      IF(KEY(NA).EQ.2)READ(5,1000)((CFORCM(I,J,NA),J=1,JMAX),I=1,IMAX)  MAI00190
      IF(KEY(NA).EQ.0) GO TO 121                                        MAI00200
      IF(KEY(NA).EQ.2) GO TO 122                                        MAI00210
      N1CNTR=N1CNTR+1                                                   MAI00220
      N1(N1CNTR)=NA                                                     MAI00230
      GO TO 121                                                         MAI00240
122   N3CNTR=N3CNTR+1                                                   MAI00250
      N3(N3CNTR)=NA                                                     MAI00260
121   CONTINUE                                                          MAI00270
      N1MAX=N1CNTR                                                      MAI00280
      N3MAX=N3CNTR                                                      MAI00290
1009  FORMAT(I10,10X,A4,10X,I6)                                         MAI00300
      READ(5,1000)$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ                  MAI00310
      READ(5,1000)AILDEF,RUDDEF,ELEDEF                                  MAI00320
1000  FORMAT(8F10.6)                                                    MAI00330
      DO 123 K2=1,N3MAX                                                 MAI00340
      DO 150 JJ=1,JMAX                                                  MAI00350
 150  P(JJ)=IDINT(B(JJ))                                                MAI00360
      I1=N3(K2)                                                         MAI00370
      WRITE(6,124 ) NAME(I1    ), (P(J),J=1,JMAX)                       MAI00380
      DO 125 K3=1,IMAX                                                  MAI00390
      A1=IDINT(A(K3))                                                   MAI00400
125   WRITE(6,126 )A1          ,(CFORCM(K3,J,I1    ),J=1,JMAX)          MAI00410
124   FORMAT('0',T55,'INPUT DATA'/////,T52,A4,'(ALPHA,BETA)'////        MAI00420
     1' ','BETA: ', 9(3X,I3,5X)///' ','ALPHA' )                        MAI00430
126   FORMAT(' ',I3,10(3X,F8.4))                                        MAI00440
 123  CONTINUE                                                          MAI00450
      M1=1                                                              MAI00460
      M2=10                                                             MAI00470
      DO 127 K4=1,3                                                     MAI00480
```

```
      IF(N1MAX.LE.1))M2=N1MAX                                                  MAI00490
      WRITE(6,130) (NAME(N1(I)),I=M1,M2)                                       MAI00500
  130 FORMAT('0',T55,'INPUT DATA' //////                                       MAI00510
     1' ','ALPHA',3X,10(A4,7X))                                                MAI00520
      DO 129 K6=1,IMAX                                                         MAI00530
      A2=IDINT(A(K6))                                                          MAI00540
  129 WRITE(6,126)A2          ,(CFORCM(K6,1, N1(I)),I=M1,M2)                   MAI00550
      M1=M1+10                                                                 MAI00560
      IF(M2.EQ.N1MAX)GO TO 128                                                 MAI00570
      M2=M2+10                                                                 MAI00580
      IF(N1MAX-M2)132,127,127                                                  MAI00590
  132 M2=N1MAX                                                                 MAI00600
  127 CONTINUE                                                                 MAI00610
  128 PI=3.141593                                                              MAI00620
      DO 426 I=1,IMAX                                                          MAI00630
  426 A(I)=A(I)*PI/180.0                                                       MAI00640
      DO 427 J=1,JMAX                                                          MAI00650
  427 B(J)=B(J)*PI/180.0                                                       MAI00660
C     DO 555 K10=1,300                                                         MAI00670
C     READ(5,1003,END=450)(R(I),I=7,12),ICCODE                                 MAI00680
C     GO TO 60                                                                 MAI00690
   70 WRITE(6,100)                                                             MAI00700
  100 FORMAT('1',T49,'INITIAL INPUT AND COMPUTED DATA'/////                    MAI00710
     1T53,'AIRCRAFT DATA'///' ',T8,'MASS',T22,'CHORD',T35,'WING AREA',T5MAI00720
     24,'IX',T69,'IY',T84,'IZ',T96,'GRAVITY',T111,'DENSITY'/                   MAI00730
     3' ',T7,'(SLUGS)',T21,'(FEET)',T35,'(FEET**2)',T50,'**********  (SMAI00740
     4LUG*FEET**2)  **********',T94,'(FT/SEC**2)',T107,'(SLUGS/FT**3)')       MAI00750
      WRITE(6,101)$MASS,CBAR,AREA,$IX,$IY,$IZ,GRAV,RHO                         MAI00760
  101 FORMAT('0',8(E13.4,2X)///)                                               MAI00770
      WRITE(6,8001)AILDEF,RUDDEF,ELEDEF                                        MAI00780
 8001 FORMAT('0',T50,'CONTROL DEFLECTIONS'/  ' ',T55,'(DEGREES)'///' ',        MAI00790
     1T36,'AILERON',T57,'RUDDER',T76,'ELEVATOR'// ' ',T37,F6.2,T57,F6.2,MAI00800
     2T76,F6.2/////)                                                           MAI00810
      READ(5,7001)M1,M2,M3,M4,M5,M6,NAB,NG,NS,M7,NSOL,NERR
      WRITE(6,7002) M1,M2,M3,M4,M5,M6,NAB,NG,NS,NSOL,NERR
      DO 7000 I1=M1,6
      F(7)=-100.0-I1*50.0
      DO 7000 I2=M2,6
      F(8)=-50.0+I2*75.0
      DO 7000 I3=M3,6
      F(9)= 0.0 +I3*10.0
      DO 7000 I4= M4,6
      F(10)=9.0-I4*10.0
      DO 7000 I5=M5,6
      F(11)=-5.0 +I5*10.0
      DO 7000 I6 =M6,9
      F(12)=5.0-I6*10.0
      DO 138 I=7,12
```

```
 138  R(I)=F(I)
      CALL EULER(ALF,BET,1)
      CALL EQNS (&1,&2,&3,&4)
      NSOL=NSOL+1
      GO TO 6999
  1   NAB=NAB +1
      GO TO 6999
 2    NG=NG+1
      GO TO 6999
 3    NS=NS+1
      GO TO 6999
 4    NERR=NERR+1
 6999 CALL EULER(ALF,BET,0)
      M7=M7+1
      IF(M7.EQ.50)WRITE(6,7002)I1,I2,I3,I4,I5,I6,NAB,NG,NS,NSOL,NERR
      IF(M7.EQ.50)M7=0
 7000 CONTINUE
      WRITE(6,7002)I1,I2,I3,I4,I5,I6,NAB,NG,NS,NSOL,NERR
      STOP
 7001 FORMAT(6I1,4X,6I10)
 7002 FORMAT(' ',6I1,4X,5I10)
      END


      SUBROUTINE EQNS (*,*,*,*)
      IMPLICIT REAL*8(A-H,O-Z,$)                                          EQN00020
      EXTERNAL FIX                                                        EQN00030
      COMMON NFLAG                                                        EQN00040
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,     EQN00050
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF
      COMMON/WORKB/ALPHA,BETA,R(12)                                       MAI00060
         DIMENSION  C(42)                                                 EQN00080
C          COMPUTATION OF THE DIRECTION CODSINES                          EQN00090
C                                                                         EQN00100
      NFLAG=1                                                             EQN00110
      A11=DCOS(R(12))*DCOS(R(11))-DSIN(R(12))*DSIN(R(11))*DCOS(R(10))     EQN00120
      A12=DCOS(R(12))*DSIN(R(11))+DSIN(R(12))*DCOS(R(11))*DCOS(R(10))     EQN00130
      A13=DSIN(R(10))*DSIN(R(12))                                         EQN00140
      A21=-DSIN(R(12))*DCOS(R(11))-DCOS(R(12))*DSIN(R(11))*DCOS(R(10))    EQN00150
      A22=-DSIN(R(12))*DSIN(R(11))+DCOS(R(12))*DCOS(R(11))*DCOS(R(10))    EQN00160
      A23=DSIN(R(10))*DCOS(R(12))                                         EQN00170
      A31=DSIN(R(10))*DSIN(R(11))                                         EQN00180
      A32=-DSIN(R(10))*DCOS(R(11))                                        EQN00190
      A33=DCOS(R(10))                                                     EQN00200
C                                                                         EQN00210
      ALPHA= DATAN((R(9)*R(8)*A32 + R(7)*A33)                             EQN00220
     1/(R(9)*R(8)*A12 + R(7)*A13))                                        EQN00230
      IF(NFLAG.EQ.0) RETURN 4
```

```
      BETA= DATAN((R(9)*R(8)*A22 + R(7)*A23)                                    EQN00240
     1/(R(9)*R(8)*A12 + R(7)*A13 ))                                             EQN00250
      IF(NFLAG.EQ.0) RETURN 4
      IF((ALPHA.GT.A(IMAX)).OR.(ALPHA.LT.A(1)).OR.(BETA.GT.B(JMAX)).OR.
     1(BETA.LT.B(1)))RETURN 1
      IF(R(9)**2*R(8)/GRAV.GT.2.0)RETURN 2
   46 DO 47 NA=1,42                                                             EQN00260
      CALL INTER(NA,C(NA))                                                      EQN00270
   47 CONTINUE                                                                  EQN00280
      VEL=DSQRT(R(7)**2 + (R(9)*R(8))**2 )                                      EQN00290
      IF(NFLAG.EQ.0) RETURN 4
      FAC13=CBAR*R(9 )*A13/2.0/VEL                                              EQN00300
      FAC23=CBAR*R(9 )*A23/2.0/VEL                                              EQN00310
      FAC33=CBAR*R(9 )*A33/2.0/VEL                                              EQN00320
      CFX=C(1) +C(25)*FAC13 + C(26)*FAC23 + C(23)*FAC33                         EQN00330
     1+C(27)*AILDEF + C(24)*RUDDEF + C( 7)*ELEDEF                               EQN00340
      CFY=C(2) +C(8 )*FAC13 + C(28)*FAC23 + C(9)*FAC33                          EQN00350
     1+C(10)*AILDEF + C(11)*RUDDEF + C(29)*ELEDEF                               EQN00360
      CFZ=C(3) +C(30)*FAC13 + C(31)*FAC23 + C(32)*FAC33                         EQN00370
     1+C(33)*AILDEF + C(34)*RUDDEF + C(12)*ELEDEF                               EQN00380
      CMX=C(4) + C(13)*FAC13+ C(35)*FAC23 + C(14)*FAC33                         EQN00390
     1+C(15)*AILDEF + C(16)*RUDDEF + C(36)*ELEDEF                               EQN00400
      CMY=C(5) +C(37)*FAC13 + C(17)*FAC23 + C(38)*FAC33                         EQN00410
     1+C(39)*AILDEF + C(40)*RUDDEF + C(18)*ELEDEF                               EQN00420
      CMZ=C(6) +C(19)*FAC13 + C(41)*FAC23 + C(20)*FAC33                         EQN00430
     1+C(21)*AILDEF + C(22)*RUDDEF + C(42)*ELEDEF                               EQN00440
   12 FAC1=((R(9)*R(8))**2 + R(7)**2 )                                          EQN00450
      IF(NFLAG.EQ.0) RETURN 4
      FAC2=RHO * AREA / 2.0/$MASS                                               EQN00460
      R(1)=-GRAV-(CFX*A13 +CFY*A23 + CFZ*A33)*FAC2*FAC1                         EQN00470
      IF(NFLAG.EQ.0) RETURN 4
      R(2)=R(9)**2*R(8)-(CFX*A11 + CFY*A21 + CFZ*A31)*FAC2*FAC1                 EQN00480
      IF(NFLAG.EQ.0) RETURN 4
  202 R(3)=CFX*A12+CFY*A22+CFZ*A32                                              EQN00490
      IF(NFLAG.EQ.0) RETURN 4
      R(4)=2.0*($IX*(DSIN(R(12))**2)+$IY*(DCOS(R(12))**2)-$IZ)*R(9)**2          EQN00500
     1*DSIN(R(10))*DCOS(R(10))+FAC1*RHO*AREA*CBAR                               EQN00510
     2*(CMX*DCOS(R(12))-CMY*DSIN(R(12)))                                        EQN00520
      IF(NFLAG.EQ.0) RETURN 4
      R(5)=CMX*A13+CMY*A23+CMZ*A33                                              EQN00530
      IF(NFLAG.EQ.0) RETURN 4
      R(6)=2.0*($IX-$IY)*R(9)**2*(DSIN(R(10))**2)*DSIN(R(12))*DCOS(R(12))       EQN00540
     1)+RHO*AREA*CBAR*FAC1*CMZ                                                  EQN00550
      IF(NFLAG.EQ.0) RETURN 4
      SUM1=DABS(R(3))+DABS(R(4))+DABS(R(5))+DABS(R(6))                          EQN00560
  400 SUM=SUM1 +DABS(R(1)) +DABS(R(2))                                          EQN00580
      IF(SUM.GT.10.0**4) RETURN 3
      CALL EULER (ALF,BET,0)
```

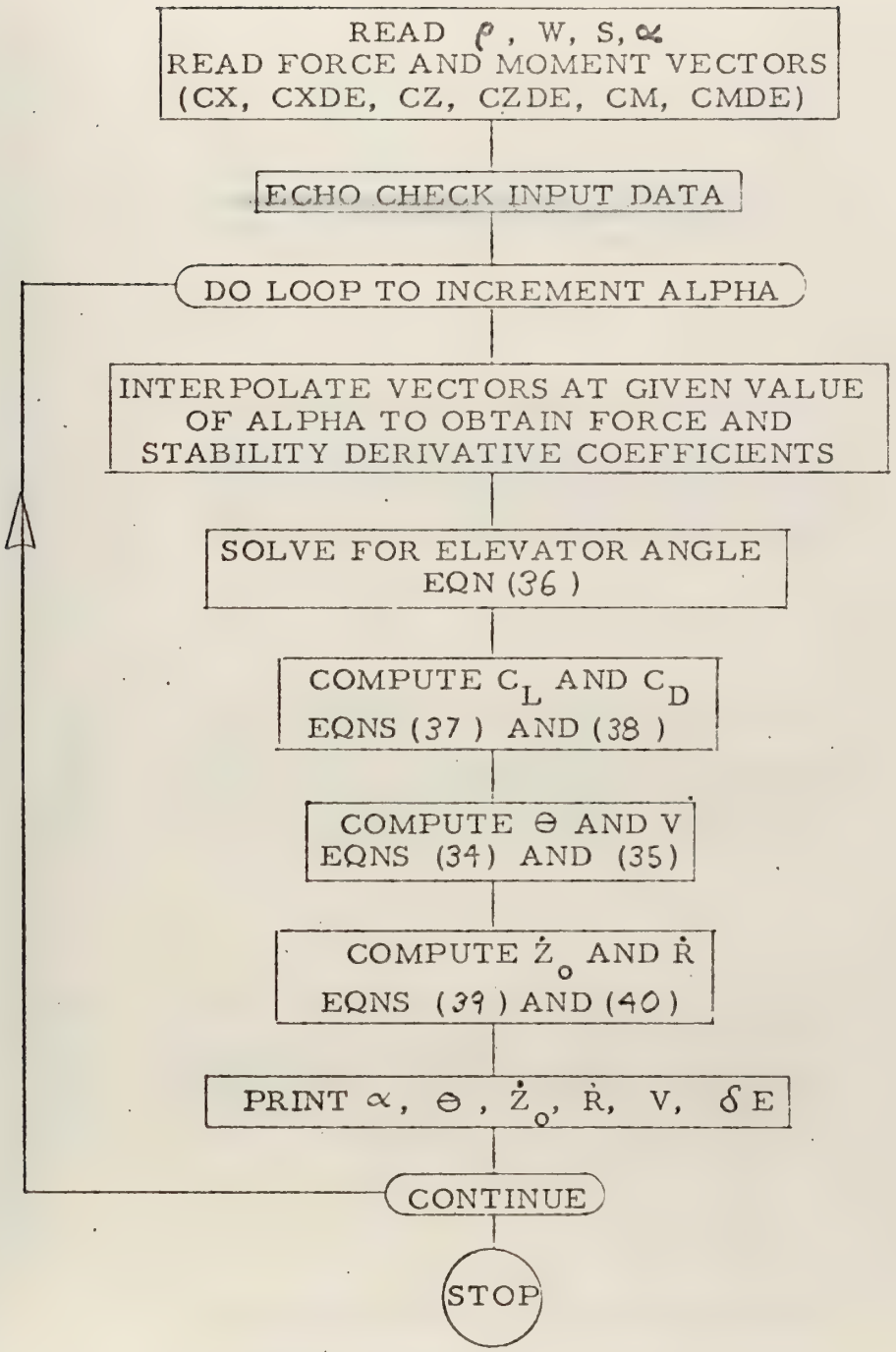
```
      WRITE(6,7003)(R(I),I=1,12),SUM
 7003 FORMAT(' ',13(E9.3,1X))
      WRITE(6,7004) ALF,BET
 7004 FORMAT(' ',2F10.2)
      CALL EULER(ALF,BET,1)
  300 RETURN                                                          EQN00600
      END                                                             EQN00610

      SUBROUTINE EULER(ALF,BET,N)                                     EUL00220
      IMPLICIT REAL *8 (A-H,O-Z,$)                                    EUL00230
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF         MAI00020
      COMMON/WORKB/ALPHA,BETA,R(12)
      PI=3.141593                                                     EUL00260
C                                                                     EUL00270
CCC       N=0:  CONVERT REFERENCE EULER ANGLES TO ORDERED EULER ANGLES EUL00280
CCC             AND THEN CONVERT ANGULAR VARIABLES FROM RADIANS TO    EUL00290
CCC             DEGREES                                               EUL00300
CCC                                                                   EUL00310
CCC       N=1:  CONVERT ANGULAR VARIABLES FROM DEGREES TO RADIANS AND EUL00320
CCC             THEN CONVERT ORDERED EULER ANGLES TO REFERENCE EULER  EUL00330
CCC             ANGLES                                                EUL00340
C                                                                     EUL00350
      IF(N.EQ.1)GO TO 2                                               EUL00360
C                                                                     EUL00370
CC        EULER ANGLE CONVERSION: REFERENCE TO ORDERED                 EUL00380
C                                                                     EUL00390
      X=R(10)                                                         EUL00400
      Y=R(11)                                                         EUL00410
      Z=R(12)                                                         EUL00420
      CTERM=DATAN(DTAN(Y/2.0)*DSIN((PI/2.0-X)/2.0)/DSIN((PI/2.0+X)/2.0))EUL00430
      DTERM=DATAN(DTAN(Y/2.0)*DCOS((PI/2.0-X)/2.0)/DCOS((PI/2.0+X)/2.0))EUL00440
      PSI=Z+CTERM+DTERM                                               EUL00450
      THETA=DTERM-CTERM                                               EUL00460
      PHI=PI/2.0-2.0*DATAN(DSIN(CTERM)/(DSIN(DTERM)*DTAN((PI/2.0-X)/2.0)EUL00470
     1))                                                              EUL00480
      R(10)=THETA*180.0/PI                                            EUL00490
      R(11)=PSI*180.0/PI                                              EUL00500
      R(12)=PHI*180.0/PI                                              EUL00510
      R(9)=R(9)*180.0/PI                                              EUL00520
      ALF=ALPHA*180.0/PI                                              EUL00530
      BET=BETA*180.0/PI                                               EUL00540
      GO TO 3                                                         EUL00550
C                                                                     EUL00560
CC        EULER ANGLE CONVERSION: ORDERED TO REFERENCE                 EUL00570
C                                                                     EUL00580
    2 DO 4 I=9,12                                                     EUL00590
```

```
 4    R(I)=R(I)*PI/180.0                                                    EUL00600
      X=R(10)                                                               EUL00610
      Y=R(11)                                                               EUL00620
      Z=R(12)                                                               EUL00630
      ATERM=DATAN(DTAN(X/2.0)*DTAN(Z/2.0))                                  EUL00640
      BTERM=DATAN(DTAN(X/2.0)/DTAN(Z/2.0))                                  EUL00650
      PHI=Y-ATERM-BTERM                                                     EUL00660
      PSI=BTERM-ATERM                                                       EUL00670
      THETA=2.0*DATAN(DSIN(ATERM)/(DSIN(BTERM)*DTAN(Z/2.0)))                EUL00680
      R(10)=THETA                                                           EUL00690
      R(11)=PSI                                                             EUL00700
      R(12)=PHI                                                             EUL00710
 3    RETURN                                                                EUL00720
      END                                                                   EUL00730


      SUBROUTINE INTER(NA,VALUE)                                            INT00010
      IMPLICIT REAL *8 (A-H,O-Z,$)
      COMMON/WORKA/$MASS,RHO,CBAR,GRAV,AREA,$IX,$IY,$IZ,PI,IMAX,JMAX,
     1A(22),B(9),CFORCM(22,9,22),KEY(42),AILDEF,RUDDEF,ELEDEF
      COMMON/WORKB/ALPHA,BETA,R(12)
      DIMENSION AMAT(22,9),YY(9),YI(22)
      II=KEY(NA)+1                                                          INT00080
C
CC        II=1; UNDEFINED STABILITY DERIVATIVE
CC        II=2; COEFFICIENT VECTOR
C         II=3; COEFFICIENT MATRIX
C
      GO TO (1,2,4 ),II
C
C         UNDEFINED STABILITY DERIVATIVES ARE EQUATED TO ZERO
C
 1    VALUE=0.0                                                             INT00100
      RETURN                                                                INT00110
C
C         VECTOR INTERPOLATION
C
 2    DO 3 I=1,IMAX                                                         INT00120
 3    YI(I)=(CFORCM(I,1,NA))
      CALL SPLIN1(A,YI,IMAX,ALPHA,VALUE)
      RETURN                                                                INT00150
C
C         MATRIX INTERPOLATION
C
 4    DO 5 I=1,IMAX                                                         INT00160
      DO 5 J= 1,JMAX                                                        INT00170
 5    AMAT(I,J)=(CFORCM(I,J,NA))
      DO 7 NR=1,IMAX                                                        INT00190
```

```
      DO 6 NC=1,JMAX                                                    INT00200
    6 YY(NC)=AMAT(NR,NC)
C
C         USE SPLIN1 TO INTERPOLATE/FILL UTILITY VECTOR (YI) WHICH
C         CORRESPONDS TO B=BETA FOR ALL VALUES OF A
C
    7 CALL SPLIN1(B,YY,JMAX,BETA,YI(NR))
C         ASSIGN VALUES
C         NOW INTERPOLATE YI AT A=ALPHA
C
      CALL SPLIN1(A,YI,IMAX,ALPHA,VALUE)                                INT00230
      RETURN                                                            INT00240
      END

      SUBROUTINE FIX(IRETCD,IERNO)
      IMPLICIT REAL*8(A-H,O-Z,$)
      COMMON NFLAG
      NFLAG=0
      IRETCD=0
      RETURN
      END
```

## APPENDIX D

### TRIM PROGRAM


READ ρ, W, S, α
READ FORCE AND MOMENT VECTORS
(CX, CXDE, CZ, CZDE, CM, CMDE)
ECHO CHECK INPUT DATA
DO LOOP TO INCREMENT ALPHA
INTERPOLATE VECTORS AT GIVEN VALUE OF ALPHA TO OBTAIN FORCE AND STABILITY DERIVATIVE COEFFICIENTS
SOLVE FOR ELEVATOR ANGLE EQN (36)
COMPUTE $C_L$ AND $C_D$ EQNS (37) AND (38)
COMPUTE Θ AND V EQNS (34) AND (35)
COMPUTE $\dot{Z}_o$ AND $\dot{R}$ EQNS (39) AND (40)
PRINT α, Θ, $\dot{Z}_o$, $\dot{R}$, V, δE
CONTINUE
STOP

Subroutines required: SPLIN1 and SPLICO

DENSITY .15530E-02 WEIGHT .50024E 05 AREA .52500E 03

| ALPHA | THETA | SINKRATE | RADIUS RATE | VELOCITY | ELEVATOR ANGLE |
|---|---|---|---|---|---|
| 0.1000E 01 | -.6122E 02 | 0.1865E 04 | 0.9824E 03 | 0.2108E 04 | 0.1453E 01 |
| 0.2000E 01 | -.1813E 02 | 0.3989E 03 | 0.1088E 04 | 0.1159E 04 | 0.8292E 00 |
| 0.3000E 01 | -.1125E 02 | 0.2135E 03 | 0.8410E 03 | 0.8676E 03 | 0.1862E 00 |
| 0.4000E 01 | -.8302E 01 | 0.1543E 03 | 0.7075E 03 | 0.7241E 03 | -.4954E 00 |
| 0.5000E 01 | -.6411E 01 | 0.1258E 03 | 0.6231E 03 | 0.6357E 03 | -.1233E 01 |
| 0.6000E 01 | -.4890E 01 | 0.1084E 03 | 0.5636E 03 | 0.5740E 03 | -.2032E 01 |
| 0.7000E 01 | -.3578E 01 | 0.9673E 02 | 0.5180E 03 | 0.5270E 03 | -.2846E 01 |
| 0.8000E 01 | -.2433E 01 | 0.8858E 02 | 0.4810E 03 | 0.4891E 03 | -.3622E 01 |
| 0.9000E 01 | -.1449E 01 | 0.8300E 02 | 0.4501E 03 | 0.4576E 03 | -.4308E 01 |
| 0.1000E 02 | -.6230E 00 | 0.7945E 02 | 0.4236E 03 | 0.4310E 03 | -.4852E 01 |
| 0.1100E 02 | 0.4709E-01 | 0.7756E 02 | 0.4008E 03 | 0.4082E 03 | -.5234E 01 |
| 0.1200E 02 | 0.5763E 00 | 0.7701E 02 | 0.3811E 03 | 0.3888E 03 | -.5534E 01 |
| 0.1300E 02 | 0.9836E 00 | 0.7750E 02 | 0.3641E 03 | 0.3723E 03 | -.5857E 01 |
| 0.1400E 02 | 0.1288E 01 | 0.7882E 02 | 0.3494E 03 | 0.3582E 03 | -.6308E 01 |
| 0.1500E 02 | 0.1509E 01 | 0.8081E 02 | 0.3368E 03 | 0.3464E 03 | -.6987E 01 |
| 0.1600E 02 | 0.1671E 01 | 0.8326E 02 | 0.3260E 03 | 0.3364E 03 | -.7929E 01 |
| 0.1700E 02 | 0.1823E 01 | 0.8575E 02 | 0.3161E 03 | 0.3275E 03 | -.8893E 01 |
| 0.1800E 02 | 0.1995E 01 | 0.8791E 02 | 0.3065E 03 | 0.3188E 03 | -.9572E 01 |
| 0.1900E 02 | 0.2200E 01 | 0.8950E 02 | 0.2964E 03 | 0.3097E 03 | -.9663E 01 |
| 0.2000E 02 | 0.2431E 01 | 0.9041E 02 | 0.2855E 03 | 0.2995E 03 | -.8875E 01 |
| 0.2100E 02 | 0.2661E 01 | 0.9074E 02 | 0.2737E 03 | 0.2884E 03 | -.7080E 01 |
| 0.2200E 02 | 0.2842E 01 | 0.9102E 02 | 0.2620E 03 | 0.2773E 03 | -.4733E 01 |
| 0.2300E 02 | 0.2950E 01 | 0.9166E 02 | 0.2512E 03 | 0.2674E 03 | -.2416E 01 |
| 0.2400E 02 | 0.2984E 01 | 0.9289E 02 | 0.2418E 03 | 0.2590E 03 | -.6701E 00 |
| 0.2500E 02 | 0.2955E 01 | 0.9483E 02 | 0.2342E 03 | 0.2528E 03 | 0.5426E-05 |
| 0.2600E 02 | 0.2883E 01 | 0.9751E 02 | 0.2284E 03 | 0.2484E 03 | -.7209E 00 |
| 0.2700E 02 | 0.2800E 01 | 0.1007E 03 | 0.2242E 03 | 0.2458E 03 | -.2548E 01 |
| 0.2800E 02 | 0.2726E 01 | 0.1043E 03 | 0.2210E 03 | 0.2444E 03 | -.5081E 01 |
| 0.2900E 02 | 0.2671E 01 | 0.1081E 03 | 0.2184E 03 | 0.2437E 03 | -.7954E 01 |
| 0.3000E 02 | 0.2636E 01 | 0.1119E 03 | 0.2162E 03 | 0.2434E 03 | -.1084E 02 |
| 0.3100E 02 | 0.2612E 01 | 0.1156E 03 | 0.2138E 03 | 0.2430E 03 | -.1342E 02 |
| 0.3200E 02 | 0.2582E 01 | 0.1189E 03 | 0.2109E 03 | 0.2421E 03 | -.1542E 02 |
| 0.3300E 02 | 0.2523E 01 | 0.1218E 03 | 0.2070E 03 | 0.2402E 03 | -.1652E 02 |
| 0.3400E 02 | 0.2409E 01 | 0.1241E 03 | 0.2018E 03 | 0.2368E 03 | -.1641E 02 |
| 0.3500E 02 | 0.2220E 01 | 0.1255E 03 | 0.1949E 03 | 0.2318E 03 | -.1471E 02 |

```
      REAL*8 CX(10),CXDE(10),CZ(10),CZDE(10),CM(10),CMDE(10),A(10),
     1ALPH,CMT,CMDET,CXTD,CXDETD,CZTD,CZDETD
      DIMENSION X(2)
      PI=3.141592
      READ(5,1000)RHO,W,S
      READ(5,1000)CX
      READ(5,1000)CXDE
      READ(5,1000)CZ
      READ(5,1000)CZDE
      READ(5,1000)CM
      READ(5,1000)CMDE
      READ(5,1000)A
      WRITE(6,2000)CX,CXDE,CZ,CZDE,CM,CMDE,A
      DO 10 I=1,10
10    A(I)=A(I)*PI/180.0
      WRITE(6,2001) RHO,W,S
2001  FORMAT('0',T10,'DENSITY',T30,'WEIGHT',T50,'AREA'/' ',3(10X,E10.5))
      WRITE(6,2003)
2003  FORMAT('0',T10,'ALPHA',T30,'THETA',T50,'SINKRATE',T70,'RADIUS RATE
     1',T90,'VELOCITY',T110,'ELEVATOR ANGLE'///)
      READ (5,1000) ALPHA
      DO 20 J=1,55
      ALPHA=ALPHA+1.0
      IF(ALPHA.GT.35.0) STOP
      ALPHA=ALPHA*PI/180.0
      ALPH =DBLE(ALPHA)
      CALL SPLIN1(A,CM,10,ALPH ,CMT)
      CALL SPLIN1(A,CMDE,10,ALPH ,CMDET)
      CALL SPLIN1(A,CX,10,ALPH ,CXTD)
      CALL SPLIN1(A,CXDE,10,ALPH ,CXDETD)
      CALL SPLIN1(A,CZ,10,ALPH ,CZTD)
      CALL SPLIN1(A,CZDE,10,ALPH ,CZDETD)
      DET=SNGL(-CMT/CMDET)
      CXT=SNGL(CXTD)
      CXDET=SNGL(CXDETD)
      CZT=SNGL(CZTD)
      CZDET=SNGL(CZDETD)
      CLT=-(CZT+CZDET*DET)*COS(ALPHA) +(CXT+CXDET*DET)*SIN(ALPHA)
      CDT=-(CXT+CXDET*DET)*COS(ALPHA)-(CZT+CZDET*DET)*SIN(ALPHA)
      X(1)=ATAN(CDT/CLT)
      X(2)=SQRT(2.0*W/(RHO*S*(CLT*COS(X(1))+CDT*SIN(X(1)))) )
      THETA=(ALPHA-X(1))*180.0/PI
      ZODOT=X(2)*SIN(X(1))
      RDOT=X(2)*COS(X(1))
      ALPHA=ALPHA*180.0/PI
      WRITE(6,2002)ALPHA,THETA,ZODOT,RDOT,X(2),DET
2002  FORMAT(' ',6(10X,E10.4))
 20   CONTINUE
```

```
2000 FORMAT('0',10E13.5)
1000 FORMAT(8F10.6)
     STOP
     END
```

## APPENDIX E

## AERODYNAMIC FORCE AND MOMENT MODELS

I. F-111 Data

| | | | |
|---|---|---|---|
| A. | Mass | 1555.0 | slugs |
| | Chord | 9.04 | feet |
| | Wing Area | 525 | $feet^2$ |
| | $I_x$ | 50,000 | slug $ft^2$ |
| | $I_y$ | 315,200 | slug $ft^2$ |
| | $I_z$ | 351,500 | slug $ft^2$ |

B. Attached data F-111 reduced to 6 x 9

II. F-4 Data

| | | | |
|---|---|---|---|
| A. | Mass | 1345 | slugs |
| | Chord | 16.03 | feet |
| | Wing Area | 530 | $feet^2$ |
| | $I_x$ | 23,000 | slug $ft^2$ |
| | $I_y$ | 134,000 | slug $ft^2$ |
| | $I_z$ | 152,000 | slug $ft^2$ |

B. Attached data (F-4) reduced to 6 x 9.

F-4

CNDR(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.000590 | -0.000530 | -0.000950 | -0.001130 | -0.001160 | -0.001130 | -0.000950 | -0.000530 | -0.000590 |
| 4 | -0.000580 | -0.000530 | -0.001100 | -0.001130 | -0.001160 | -0.001130 | -0.001100 | -0.000530 | -0.000580 |
| 9 | -0.000590 | -0.000530 | -0.001040 | -0.001130 | -0.001130 | -0.001130 | -0.001040 | -0.000530 | -0.000590 |
| 14 | -0.000580 | -0.000520 | -0.001300 | -0.001120 | -0.001050 | -0.001120 | -0.001300 | -0.000520 | -0.000580 |
| 19 | -0.000560 | -0.000510 | -0.000880 | -0.000730 | -0.000910 | -0.000730 | -0.000880 | -0.000510 | -0.000560 |
| 24 | -0.000460 | -0.000460 | -0.000570 | -0.000450 | -0.000750 | -0.000450 | -0.000570 | -0.000460 | -0.000460 |
| 29 | -0.000440 | -0.000450 | -0.000400 | -0.000280 | -0.000520 | -0.000280 | -0.000400 | -0.000450 | -0.000440 |
| 34 | -0.000420 | -0.000430 | -0.000270 | -0.000210 | -0.000360 | -0.000210 | -0.000270 | -0.000430 | -0.000420 |
| 39 | -0.000420 | -0.000390 | -0.000220 | -0.000170 | -0.000220 | -0.000170 | -0.000220 | -0.000390 | -0.000420 |
| 44 | -0.000380 | -0.000220 | -0.000140 | -0.000150 | -0.000040 | -0.000150 | -0.000140 | -0.000220 | -0.000380 |
| 49 | -0.000290 | -0.000150 | -0.000150 | -0.000120 | 0.0 | -0.000120 | -0.000150 | -0.000150 | -0.000290 |
| 54 | -0.000200 | -0.000140 | -0.000160 | -0.000110 | 0.0 | -0.000110 | -0.000160 | -0.000140 | -0.000200 |
| 59 | -0.000070 | -0.000150 | -0.000130 | -0.000090 | 0.0 | -0.000090 | -0.000130 | -0.000150 | -0.000070 |
| 64 | -0.000050 | -0.000130 | -0.000120 | 0.000085 | 0.0 | 0.000085 | -0.000120 | -0.000130 | -0.000050 |
| 69 | -0.000110 | -0.000090 | -0.000130 | -0.000080 | 0.0 | -0.000080 | -0.000130 | -0.000090 | -0.000110 |
| 74 | -0.000160 | -0.000070 | -0.000130 | 0.000070 | 0.0 | 0.000070 | -0.000130 | -0.000070 | -0.000160 |
| 79 | -0.000180 | -0.000100 | -0.000140 | -0.000080 | 0.0 | -0.000080 | -0.000140 | -0.000100 | -0.000180 |
| 84 | -0.000190 | -0.000150 | -0.000160 | 0.000080 | 0.0 | 0.000080 | -0.000160 | -0.000150 | -0.000190 |
| 89 | -0.000210 | -0.000190 | -0.000180 | -0.000100 | 0.0 | -0.000100 | -0.000180 | -0.000190 | -0.000210 |

F-4

CNDA(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.000180 | 0.000050 | 0.000070 | -0.000020 | 0.0 | -0.000020 | 0.000070 | 0.000050 | 0.000180 |
| 4 | 0.000110 | 0.000060 | 0.000070 | -0.000010 | 0.0 | -0.000010 | 0.000070 | 0.000060 | 0.000110 |
| 9 | 0.000110 | 0.000070 | 0.000080 | 0.000030 | 0.000010 | 0.000030 | 0.000080 | 0.000070 | 0.000110 |
| 14 | 0.000080 | 0.000090 | 0.000080 | 0.000070 | 0.000070 | 0.000070 | 0.000080 | 0.000090 | 0.000080 |
| 19 | 0.000050 | 0.000070 | 0.000120 | 0.000120 | 0.000160 | 0.000120 | 0.000120 | 0.000070 | 0.000050 |
| 24 | 0.000010 | 0.000030 | 0.000130 | 0.000150 | 0.000150 | 0.000150 | 0.000130 | 0.000030 | 0.000010 |
| 29 | -0.000030 | 0.0 | 0.000130 | 0.000160 | 0.000150 | 0.000160 | 0.000130 | 0.0 | -0.000030 |
| 34 | -0.000020 | -0.000010 | 0.000130 | 0.000160 | 0.000130 | 0.000160 | 0.000130 | -0.000010 | -0.000020 |
| 39 | -0.000030 | 0.0 | 0.000140 | 0.000170 | 0.000200 | 0.000170 | 0.000140 | 0.0 | -0.000030 |
| 44 | 0.000010 | 0.000020 | 0.000140 | 0.000160 | 0.000170 | 0.000160 | 0.000140 | 0.000020 | 0.000010 |
| 49 | 0.000020 | 0.000030 | 0.000150 | 0.000140 | 0.000200 | 0.000140 | 0.000150 | 0.000030 | 0.000020 |
| 54 | 0.000040 | 0.000040 | 0.000160 | 0.000110 | 0.000170 | 0.000110 | 0.000160 | 0.000040 | 0.000040 |
| 59 | 0.000050 | 0.000060 | 0.000160 | 0.000130 | 0.000170 | 0.000130 | 0.000160 | 0.000060 | 0.000050 |
| 64 | 0.000060 | 0.000080 | 0.000160 | 0.000190 | 0.000200 | 0.000190 | 0.000160 | 0.000080 | 0.000060 |
| 69 | 0.000080 | 0.000100 | 0.000160 | 0.000240 | 0.000160 | 0.000240 | 0.000160 | 0.000100 | 0.000080 |
| 74 | 0.000100 | 0.000120 | 0.000160 | 0.000260 | 0.000130 | 0.000260 | 0.000160 | 0.000120 | 0.000100 |
| 79 | 0.000130 | 0.000150 | 0.000160 | 0.000270 | 0.000150 | 0.000270 | 0.000160 | 0.000150 | 0.000130 |
| 84 | 0.000150 | 0.000170 | 0.000150 | 0.000250 | 0.000170 | 0.000250 | 0.000150 | 0.000170 | 0.000150 |
| 89 | 0.000160 | 0.000180 | 0.000140 | 0.000230 | 0.000150 | 0.000230 | 0.000140 | 0.000180 | 0.000160 |

CN (ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.238800 | -0.202600 | -0.149500 | -0.058610 | 0.0 | 0.058610 | 0.149500 | 0.202600 | 0.238800 |
| 4 | -0.242400 | -0.209800 | -0.152000 | -0.062710 | 0.0 | 0.062710 | 0.152000 | 0.209800 | 0.242400 |
| 9 | -0.242400 | -0.211100 | -0.149500 | -0.053550 | 0.0 | 0.053550 | 0.149500 | 0.211100 | 0.242400 |
| 14 | -0.241200 | -0.207400 | -0.141600 | -0.050650 | 0.0 | 0.050650 | 0.141600 | 0.207400 | 0.241200 |
| 19 | -0.123000 | -0.097690 | -0.078390 | -0.015680 | 0.0 | 0.015680 | 0.078390 | 0.097690 | 0.123000 |
| 24 | -0.061510 | -0.001206 | 0.030150 | 0.065130 | 0.0 | -0.065130 | -0.030150 | 0.001206 | 0.061510 |
| 29 | -0.053070 | 0.033770 | 0.112900 | 0.090450 | 0.0 | -0.090450 | -0.112900 | -0.033770 | 0.053070 |
| 34 | -0.045830 | 0.067540 | 0.099380 | 0.094070 | 0.0 | -0.094070 | -0.099380 | -0.067540 | 0.045830 |
| 39 | -0.018090 | 0.078390 | 0.103700 | 0.052580 | 0.0 | -0.052580 | -0.103700 | -0.078390 | 0.018090 |
| 44 | -0.008442 | 0.086830 | 0.080800 | 0.047030 | 0.0 | -0.047030 | -0.080800 | -0.086830 | 0.008442 |
| 49 | -0.033770 | 0.089250 | 0.033770 | 0.060300 | 0.0 | -0.060300 | -0.033770 | -0.089250 | 0.033770 |
| 54 | -0.007236 | 0.089250 | 0.115800 | 0.106100 | 0.0 | -0.106100 | -0.115800 | -0.089250 | 0.007236 |
| 59 | -0.006030 | 0.060300 | 0.094070 | 0.089250 | 0.0 | -0.089250 | -0.094070 | -0.060300 | 0.006030 |
| 64 | -0.008442 | 0.038590 | 0.107300 | 0.125400 | 0.0 | -0.125400 | -0.107300 | -0.038590 | 0.008442 |
| 69 | -0.031360 | 0.028940 | 0.068740 | 0.067540 | 0.0 | -0.067540 | -0.068740 | -0.028940 | 0.031360 |
| 74 | -0.079600 | 0.014470 | 0.042210 | 0.048240 | 0.0 | -0.048240 | -0.042210 | -0.014470 | 0.079600 |
| 79 | -0.095280 | 0.0 | 0.037390 | 0.039800 | 0.0 | -0.039800 | -0.037390 | 0.0 | 0.095280 |
| 84 | -0.144700 | -0.048240 | 0.003618 | 0.012060 | 0.0 | -0.012060 | -0.003618 | 0.048240 | 0.144700 |
| 89 | -0.173700 | -0.084420 | -0.026530 | 0.006030 | 0.0 | -0.006030 | 0.026530 | 0.084420 | 0.173700 |



CMDS(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.007200 | -0.009800 | -0.009000 | -0.009800 | -0.010200 | -0.009800 | -0.009000 | -0.009800 | -0.007200 |
| 4 | -0.007800 | -0.010200 | -0.009700 | -0.010300 | -0.009900 | -0.010300 | -0.009700 | -0.010200 | -0.007800 |
| 9 | -0.007900 | -0.010200 | -0.010200 | -0.010200 | -0.010000 | -0.010200 | -0.010200 | -0.010200 | -0.007900 |
| 14 | -0.007400 | -0.009800 | -0.010300 | -0.009800 | -0.010100 | -0.009800 | -0.010300 | -0.009800 | -0.007400 |
| 19 | -0.006500 | -0.009000 | -0.010000 | -0.009000 | -0.009600 | -0.009000 | -0.010000 | -0.009000 | -0.006500 |
| 24 | -0.005800 | -0.007700 | -0.008300 | -0.007600 | -0.008700 | -0.007600 | -0.008300 | -0.007700 | -0.005800 |
| 29 | -0.005300 | -0.006000 | -0.006600 | -0.006100 | -0.007000 | -0.006100 | -0.006600 | -0.006000 | -0.005300 |
| 34 | -0.005000 | -0.004200 | -0.005200 | -0.005200 | -0.004900 | -0.005200 | -0.005200 | -0.004200 | -0.005000 |
| 39 | -0.004700 | -0.003400 | -0.004400 | -0.004400 | -0.003000 | -0.004400 | -0.004400 | -0.003400 | -0.004700 |
| 44 | -0.004200 | -0.003200 | -0.004000 | -0.004000 | -0.001900 | -0.004000 | -0.004000 | -0.003200 | -0.004200 |
| 49 | -0.003600 | -0.003400 | -0.004400 | -0.003700 | -0.001600 | -0.003700 | -0.004400 | -0.003400 | -0.003600 |
| 54 | -0.003000 | -0.004400 | -0.004900 | -0.003600 | -0.002000 | -0.003600 | -0.004900 | -0.004400 | -0.003000 |
| 59 | -0.002400 | -0.006000 | -0.005000 | -0.003600 | -0.002700 | -0.003600 | -0.005000 | -0.006000 | -0.002400 |
| 64 | -0.003300 | -0.005200 | -0.004800 | -0.004000 | -0.003200 | -0.004000 | -0.004800 | -0.005200 | -0.003300 |
| 69 | -0.004000 | -0.004000 | -0.004500 | -0.004300 | -0.003500 | -0.004300 | -0.004500 | -0.004000 | -0.004000 |
| 74 | -0.004100 | -0.003300 | -0.004200 | -0.004500 | -0.003800 | -0.004500 | -0.004200 | -0.003300 | -0.004100 |
| 79 | -0.004200 | -0.003600 | -0.003800 | -0.003600 | -0.004200 | -0.003600 | -0.003800 | -0.003600 | -0.004200 |
| 84 | -0.004500 | -0.003700 | -0.003100 | -0.002700 | -0.004600 | -0.002700 | -0.003100 | -0.003700 | -0.004500 |
| 89 | -0.005400 | -0.003300 | -0.001700 | -0.002600 | -0.003600 | -0.002600 | -0.001700 | -0.003300 | -0.005400 |

CZ (ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.250000 | 0.075000 | -0.120000 | -0.120000 | -0.120000 | -0.120000 | -0.120000 | 0.075000 | 0.250000 |
| 4 | -0.050000 | -0.210000 | -0.370000 | -0.370000 | -0.370000 | -0.370000 | -0.370000 | -0.210000 | -0.050000 |
| 9 | -0.330000 | 0.480000 | -0.630000 | -0.630000 | -0.630000 | -0.630000 | -0.630000 | 0.480000 | -0.330000 |
| 14 | -0.570000 | -0.720000 | -0.860000 | -0.860000 | -0.860000 | -0.860000 | -0.860000 | -0.720000 | -0.570000 |
| 19 | -0.780000 | -0.940000 | -1.330000 | -1.330000 | -1.330000 | -1.330000 | -1.330000 | -0.940000 | -0.780000 |
| 24 | -0.940000 | -1.060000 | -1.190000 | -1.190000 | -1.190000 | -1.190000 | -1.190000 | -1.060000 | -0.940000 |
| 29 | -1.030000 | -1.150000 | -1.260000 | -1.260000 | -1.260000 | -1.260000 | -1.260000 | -1.150000 | -1.030000 |
| 34 | -1.120000 | -1.220000 | -1.320000 | -1.320000 | -1.320000 | -1.320000 | -1.320000 | -1.220000 | -1.120000 |
| 39 | -1.190000 | -1.290000 | -1.390000 | -1.390000 | -1.390000 | -1.390000 | -1.390000 | -1.290000 | -1.190000 |
| 44 | -1.250000 | -1.350000 | -1.460000 | -1.460000 | -1.460000 | -1.460000 | -1.460000 | -1.350000 | -1.250000 |
| 49 | -1.330000 | -1.430000 | -1.530000 | -1.530000 | -1.530000 | -1.530000 | -1.530000 | -1.430000 | -1.330000 |
| 54 | -1.400000 | -1.500000 | -1.600000 | -1.600000 | -1.600000 | -1.600000 | -1.600000 | -1.500000 | -1.400000 |
| 59 | -1.460000 | -1.550000 | -1.640000 | -1.640000 | -1.640000 | -1.640000 | -1.640000 | -1.550000 | -1.460000 |
| 64 | -1.510000 | -1.590000 | -1.670000 | -1.670000 | -1.670000 | -1.670000 | -1.670000 | -1.590000 | -1.510000 |
| 69 | -1.550000 | -1.620000 | -1.690000 | -1.690000 | -1.690000 | -1.690000 | -1.690000 | -1.620000 | -1.550000 |
| 74 | -1.580000 | -1.640000 | -1.700000 | -1.700000 | -1.700000 | -1.700000 | -1.700000 | -1.640000 | -1.580000 |
| 79 | -1.570000 | -1.640000 | -1.700000 | -1.700000 | -1.700000 | -1.700000 | -1.700000 | -1.640000 | -1.570000 |
| 84 | -1.540000 | -1.640000 | -1.700000 | -1.700000 | -1.700000 | -1.700000 | -1.700000 | -1.640000 | -1.540000 |
| 89 | -1.560000 | -1.620000 | -1.680000 | -1.680000 | -1.680000 | -1.680000 | -1.680000 | -1.620000 | -1.560000 |





CM (ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.100000 | 0.035000 | -0.030000 | -0.015000 | 0.0 | -0.015000 | -0.030000 | 0.035000 | 0.100000 |
| 4 | 0.095000 | 0.030000 | -0.035000 | -0.020000 | -0.005000 | -0.020000 | -0.035000 | 0.030000 | 0.095000 |
| 9 | 0.090000 | 0.025000 | -0.040000 | -0.025000 | -0.010000 | -0.025000 | -0.040000 | 0.025000 | 0.090000 |
| 14 | 0.075000 | 0.011000 | -0.053000 | -0.039000 | -0.025000 | -0.039000 | -0.053000 | 0.011000 | 0.075000 |
| 19 | 0.075000 | 0.010000 | -0.055000 | -0.040000 | -0.025000 | -0.040000 | -0.055000 | 0.010000 | 0.075000 |
| 24 | 0.051000 | -0.014000 | -0.079000 | -0.065000 | -0.050000 | -0.065000 | -0.079000 | -0.014000 | 0.051000 |
| 29 | 0.032000 | -0.034000 | -0.100000 | -0.086000 | -0.072000 | -0.086000 | -0.100000 | -0.034000 | 0.032000 |
| 34 | 0.016000 | -0.049000 | -0.114000 | -0.100000 | -0.085000 | -0.100000 | -0.114000 | -0.049000 | 0.016000 |
| 39 | 0.010000 | -0.055000 | -0.121000 | -0.106000 | -0.091000 | -0.106000 | -0.121000 | -0.055000 | 0.010000 |
| 44 | 0.005000 | -0.070000 | -0.135000 | -0.120000 | -0.105000 | -0.120000 | -0.135000 | -0.070000 | 0.005000 |
| 49 | -0.032000 | -0.097000 | -0.161000 | -0.146000 | -0.131000 | -0.146000 | -0.161000 | -0.097000 | -0.032000 |
| 54 | -0.062000 | -0.127000 | -0.191000 | -0.176000 | -0.161000 | -0.176000 | -0.191000 | -0.127000 | -0.062000 |
| 59 | -0.090000 | -0.155000 | -0.220000 | -0.205000 | -0.190000 | -0.205000 | -0.220000 | -0.155000 | -0.090000 |
| 64 | -0.120000 | -0.185000 | -0.250000 | -0.235000 | -0.220000 | -0.235000 | -0.250000 | -0.185000 | -0.120000 |
| 69 | -0.154000 | -0.219000 | -0.285000 | -0.270000 | -0.255000 | -0.270000 | -0.285000 | -0.219000 | -0.154000 |
| 74 | -0.194000 | -0.259000 | -0.325000 | -0.310000 | -0.295000 | -0.310000 | -0.325000 | -0.259000 | -0.194000 |
| 79 | -0.225000 | -0.291000 | -0.357000 | -0.341000 | -0.325000 | -0.341000 | -0.357000 | -0.291000 | -0.225000 |
| 84 | -0.260000 | -0.325000 | -0.390000 | -0.375000 | -0.360000 | -0.375000 | -0.390000 | -0.325000 | -0.260000 |
| 89 | -0.275000 | -0.340000 | -0.404000 | -0.390000 | -0.375000 | -0.390000 | -0.404000 | -0.340000 | -0.275000 |

F-4

CY (ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.575000 | 0.457000 | 0.300000 | 0.129000 | 0.0 | -0.129000 | -0.300000 | -0.457000 | -0.575000 |
| 4 | 0.565000 | 0.450000 | 0.295000 | 0.125000 | 0.0 | -0.125000 | -0.295000 | -0.450000 | -0.565000 |
| 9 | 0.560000 | 0.450000 | 0.300000 | 0.125000 | 0.0 | -0.125000 | -0.300000 | -0.450000 | -0.560000 |
| 14 | 0.555000 | 0.430000 | 0.290000 | 0.120000 | 0.0 | -0.120000 | -0.290000 | -0.430000 | -0.555000 |
| 19 | 0.555000 | 0.408000 | 0.285000 | 0.110000 | 0.0 | -0.110000 | -0.285000 | -0.408000 | -0.555000 |
| 24 | 0.485000 | 0.335000 | 0.205000 | 0.053000 | 0.0 | -0.053000 | -0.205000 | -0.335000 | -0.485000 |
| 29 | 0.445000 | 0.335000 | 0.180000 | 0.050000 | 0.0 | -0.050000 | -0.180000 | -0.335000 | -0.445000 |
| 34 | 0.420000 | 0.295000 | 0.135000 | 0.065000 | 0.0 | -0.065000 | -0.135000 | -0.295000 | -0.420000 |
| 39 | 0.390000 | 0.235000 | 0.120000 | 0.083000 | 0.0 | -0.083000 | -0.120000 | -0.235000 | -0.390000 |
| 44 | 0.375000 | 0.208000 | 0.135000 | 0.105000 | 0.0 | -0.105000 | -0.135000 | -0.208000 | -0.375000 |
| 49 | 0.300000 | 0.145000 | 0.140000 | 0.065000 | 0.0 | -0.065000 | -0.140000 | -0.145000 | -0.300000 |
| 54 | 0.245000 | 0.125000 | 0.088000 | 0.076000 | 0.0 | -0.076000 | -0.088000 | -0.125000 | -0.245000 |
| 59 | 0.205000 | 0.070000 | 0.020000 | 0.090000 | 0.0 | -0.090000 | -0.020000 | -0.070000 | -0.205000 |
| 64 | 0.170000 | 0.050000 | -0.010000 | 0.040000 | 0.0 | -0.040000 | 0.010000 | -0.050000 | -0.170000 |
| 69 | 0.185000 | 0.065000 | 0.020000 | 0.035000 | 0.0 | -0.035000 | -0.020000 | -0.065000 | -0.185000 |
| 74 | 0.185000 | 0.090000 | 0.025000 | 0.010000 | 0.0 | -0.010000 | -0.025000 | -0.090000 | -0.185000 |
| 79 | 0.195000 | 0.095000 | 0.020000 | -0.030000 | 0.0 | 0.030000 | -0.020000 | -0.095000 | -0.195000 |
| 84 | 0.225000 | 0.130000 | 0.025000 | -0.015000 | 0.0 | 0.015000 | -0.025000 | -0.130000 | -0.225000 |
| 89 | 0.235000 | 0.150000 | 0.050000 | -0.010000 | 0.0 | 0.010000 | -0.050000 | -0.150000 | -0.235000 |

F-4

CL (ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.028940 | -0.029430 | -0.031360 | -0.024120 | 0.0 | -0.024120 | -0.031360 | -0.029430 | -0.028940 |
| 4 | -0.091660 | -0.084420 | -0.077190 | -0.060300 | 0.0 | -0.060300 | -0.077190 | -0.084420 | -0.091660 |
| 9 | -0.082010 | -0.089250 | -0.086830 | -0.057890 | 0.0 | -0.057890 | -0.086830 | -0.089250 | -0.082010 |
| 14 | -0.084420 | -0.077190 | -0.067540 | -0.043420 | 0.0 | -0.043420 | -0.067540 | -0.077190 | -0.084420 |
| 19 | -0.089250 | -0.067540 | -0.055480 | -0.012060 | 0.0 | -0.012060 | -0.055480 | -0.067540 | -0.089250 |
| 24 | -0.098890 | -0.079600 | -0.033770 | -0.007236 | 0.0 | -0.007236 | -0.033770 | -0.079600 | -0.098890 |
| 29 | -0.166400 | -0.094070 | -0.028940 | -0.009648 | 0.0 | -0.009648 | -0.028940 | -0.094070 | -0.166400 |
| 34 | -0.214700 | -0.118200 | -0.074770 | -0.004824 | 0.0 | -0.004824 | -0.074770 | -0.118200 | -0.214700 |
| 39 | -0.224300 | -0.139900 | -0.086830 | -0.041000 | 0.0 | -0.041000 | -0.086830 | -0.139900 | -0.224300 |
| 44 | -0.234000 | -0.152000 | -0.108500 | -0.048240 | 0.0 | -0.048240 | -0.108500 | -0.152000 | -0.234000 |
| 49 | -0.224300 | -0.166400 | -0.103700 | -0.045830 | 0.0 | -0.045830 | -0.103700 | -0.166400 | -0.224300 |
| 54 | -0.219500 | -0.173700 | -0.111000 | -0.053070 | 0.0 | -0.053070 | -0.111000 | -0.173700 | -0.219500 |
| 59 | -0.221900 | -0.176100 | -0.118200 | -0.057890 | 0.0 | -0.057890 | -0.118200 | -0.176100 | -0.221900 |
| 64 | -0.231600 | -0.178500 | -0.123000 | -0.062710 | 0.0 | -0.062710 | -0.123000 | -0.178500 | -0.231600 |
| 69 | -0.236400 | -0.178500 | -0.120600 | -0.067540 | 0.0 | -0.067540 | -0.120600 | -0.178500 | -0.236400 |
| 74 | -0.236400 | -0.180900 | -0.120600 | -0.065130 | 0.0 | -0.065130 | -0.120600 | -0.180900 | -0.236400 |
| 79 | -0.238800 | -0.188100 | -0.125400 | -0.065130 | 0.0 | -0.065130 | -0.125400 | -0.188100 | -0.238800 |
| 84 | -0.241200 | -0.195400 | -0.132700 | -0.074770 | 0.0 | -0.074770 | -0.132700 | -0.195400 | -0.241200 |
| 89 | -0.246000 | -0.197800 | -0.142300 | -0.065130 | 0.0 | -0.065130 | -0.142300 | -0.197800 | -0.246000 |

F-4

CYDR(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.001670 | 0.001670 | 0.001670 | 0.001670 | 0.001670 | 0.001670 | 0.001670 | 0.001670 | 0.001670 |
| 4 | 0.001690 | 0.001690 | 0.001690 | 0.001690 | 0.001690 | 0.001690 | 0.001690 | 0.001690 | 0.001690 |
| 9 | 0.001660 | 0.001660 | 0.001660 | 0.001660 | 0.001660 | 0.001660 | 0.001660 | 0.001660 | 0.001660 |
| 14 | 0.001600 | 0.001600 | 0.001600 | 0.001600 | 0.001600 | 0.001600 | 0.001600 | 0.001600 | 0.001600 |
| 19 | 0.001490 | 0.001490 | 0.001490 | 0.001490 | 0.001490 | 0.001490 | 0.001490 | 0.001490 | 0.001490 |
| 24 | 0.000960 | 0.000960 | 0.001220 | 0.001220 | 0.001390 | 0.001220 | 0.001220 | 0.000960 | 0.000960 |
| 29 | 0.000590 | 0.000590 | 0.000880 | 0.000880 | 0.001240 | 0.000880 | 0.000880 | 0.000590 | 0.000590 |
| 34 | 0.000500 | 0.000500 | 0.000570 | 0.000570 | 0.001010 | 0.000570 | 0.000570 | 0.000500 | 0.000500 |
| 39 | 0.000410 | 0.000410 | 0.000230 | 0.000230 | 0.000440 | 0.000230 | 0.000230 | 0.000410 | 0.000410 |
| 44 | 0.000200 | 0.000200 | 0.000060 | 0.000006 | 0.000060 | 0.000006 | 0.000060 | 0.000200 | 0.000200 |
| 49 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 54 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 59 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 64 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 69 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 74 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 79 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 84 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 89 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

F-4

CX (ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.008500 | -0.014000 | -0.024500 | -0.029000 | -0.035500 | -0.029000 | -0.024500 | -0.014000 | -0.008500 |
| 4 | -0.003500 | -0.010000 | -0.015000 | -0.015000 | -0.020000 | -0.015000 | -0.015000 | -0.010000 | -0.003500 |
| 9 | 0.003500 | -0.003500 | -0.002500 | 0.002500 | 0.002000 | 0.002500 | -0.002500 | -0.003500 | 0.003500 |
| 14 | 0.008500 | 0.003000 | 0.006500 | 0.009000 | 0.009000 | 0.009000 | 0.006500 | 0.003000 | 0.008500 |
| 19 | 0.015500 | 0.009500 | 0.009000 | -0.006500 | -0.015000 | -0.006500 | 0.009000 | 0.009500 | 0.015500 |
| 24 | 0.021000 | 0.013500 | -0.000500 | -0.026500 | -0.033500 | -0.026500 | -0.000500 | 0.013500 | 0.021000 |
| 29 | 0.019000 | 0.013000 | -0.016000 | -0.030500 | -0.031000 | -0.030500 | -0.016000 | 0.013000 | 0.019000 |
| 34 | 0.018000 | 0.006000 | -0.024000 | -0.029000 | -0.030500 | -0.029000 | -0.024000 | 0.006000 | 0.018000 |
| 39 | 0.022000 | -0.001000 | -0.021000 | -0.024000 | -0.032500 | -0.024000 | -0.021000 | -0.001000 | 0.022000 |
| 44 | 0.021500 | -0.003000 | -0.012000 | -0.015500 | -0.021500 | -0.015500 | -0.012000 | -0.003000 | 0.021500 |
| 49 | 0.008500 | 0.000500 | 0.004500 | -0.002500 | -0.007500 | -0.002500 | 0.004500 | 0.000500 | 0.008500 |
| 54 | 0.009500 | 0.007500 | 0.013500 | 0.010000 | 0.011500 | 0.010000 | 0.013500 | 0.007500 | 0.009500 |
| 59 | 0.015000 | 0.016000 | 0.018500 | 0.020500 | 0.030000 | 0.020500 | 0.018500 | 0.016000 | 0.015000 |
| 64 | 0.021500 | 0.023500 | 0.024000 | 0.029000 | 0.043000 | 0.029000 | 0.024000 | 0.023500 | 0.021500 |
| 69 | 0.030500 | 0.027500 | 0.025500 | 0.026000 | 0.048000 | 0.026000 | 0.025500 | 0.027500 | 0.030500 |
| 74 | 0.039500 | 0.025500 | 0.024000 | 0.033000 | 0.047000 | 0.033000 | 0.024000 | 0.025500 | 0.039500 |
| 79 | 0.045500 | 0.033500 | 0.026000 | 0.030500 | 0.040000 | 0.030500 | 0.026000 | 0.033500 | 0.045500 |
| 84 | 0.039500 | 0.027500 | 0.025500 | 0.027000 | 0.025000 | 0.027000 | 0.025500 | 0.027500 | 0.039500 |
| 89 | 0.028500 | 0.029000 | 0.024000 | 0.039000 | -0.005000 | 0.039000 | 0.024000 | 0.039000 | 0.028500 |

F-4

CZDS(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.007000 | -0.007550 | -0.008100 | -0.008000 | -0.007900 | -0.008000 | -0.008100 | -0.007550 | -0.007000 |
| 4 | -0.006700 | -0.006000 | -0.008100 | -0.007850 | -0.007600 | -0.007850 | -0.008100 | -0.006000 | -0.006700 |
| 9 | -0.006300 | -0.007100 | -0.007900 | -0.007600 | -0.007300 | -0.007600 | -0.007900 | -0.007100 | -0.006300 |
| 14 | -0.005700 | -0.006500 | -0.007300 | -0.007200 | -0.007100 | -0.007200 | -0.007300 | -0.006500 | -0.005700 |
| 19 | -0.004800 | -0.005800 | -0.006700 | -0.006900 | -0.007100 | -0.006900 | -0.006700 | -0.005800 | -0.004800 |
| 24 | -0.003900 | -0.004950 | -0.006000 | -0.006700 | -0.007400 | -0.006700 | -0.006000 | -0.004950 | -0.003900 |
| 29 | -0.003600 | -0.004500 | -0.005400 | -0.006600 | -0.007700 | -0.006600 | -0.005400 | -0.004500 | -0.003600 |
| 34 | -0.003800 | -0.004350 | -0.004900 | -0.005900 | -0.006800 | -0.005900 | -0.004900 | -0.004350 | -0.003800 |
| 39 | -0.003300 | -0.003900 | -0.004500 | -0.004650 | -0.004800 | -0.004650 | -0.004500 | -0.003900 | -0.003300 |
| 44 | -0.002600 | -0.003250 | -0.003900 | -0.004000 | -0.004100 | -0.004000 | -0.003900 | -0.003250 | -0.002600 |
| 49 | -0.001800 | -0.002600 | -0.003400 | -0.003570 | -0.003750 | -0.003570 | -0.003400 | -0.002600 | -0.001800 |
| 54 | -0.001200 | -0.001950 | -0.002700 | -0.003000 | -0.003300 | -0.003000 | -0.002700 | -0.001950 | -0.001200 |
| 59 | -0.001450 | -0.001850 | -0.002250 | -0.002370 | -0.002500 | -0.002370 | -0.002250 | -0.001850 | -0.001450 |
| 64 | -0.002600 | -0.002350 | -0.002100 | -0.001950 | -0.001800 | -0.001950 | -0.002100 | -0.002350 | -0.002600 |
| 69 | -0.002350 | -0.002620 | -0.002900 | -0.002150 | -0.001400 | -0.002150 | -0.002900 | -0.002620 | -0.002350 |
| 74 | -0.001200 | -0.001950 | -0.002700 | -0.002050 | -0.001400 | -0.002050 | -0.002700 | -0.001950 | -0.001200 |
| 79 | -0.001000 | -0.001950 | -0.002900 | -0.002250 | -0.001600 | -0.002250 | -0.002900 | -0.001950 | -0.001000 |
| 84 | -0.000600 | -0.001700 | -0.002800 | -0.002400 | -0.002000 | -0.002400 | -0.002800 | -0.001700 | -0.000600 |
| 89 | -0.001700 | -0.002100 | 0.002550 | -0.002520 | -0.002400 | -0.002520 | 0.002550 | -0.002100 | -0.001700 |

F-4

CXDS(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.001530 | 0.001530 | 0.001530 | 0.001600 | 0.001600 | 0.001600 | 0.001530 | 0.001530 | 0.001530 |
| 4 | -0.001520 | 0.001520 | 0.001520 | 0.001600 | 0.001600 | 0.001600 | 0.001520 | 0.001520 | -0.001520 |
| 9 | 0.001300 | 0.001300 | 0.001300 | 0.001450 | 0.001450 | 0.001450 | 0.001300 | 0.001300 | 0.001300 |
| 14 | -0.000990 | 0.000990 | 0.000990 | 0.001150 | 0.001150 | 0.001150 | 0.000990 | 0.000990 | -0.000990 |
| 19 | 0.000500 | 0.000500 | 0.000500 | 0.000750 | 0.000750 | 0.000750 | 0.000500 | 0.000500 | 0.000500 |
| 24 | 0.000100 | 0.000100 | 0.000100 | 0.000300 | 0.000300 | 0.000300 | 0.000100 | 0.000100 | 0.000100 |
| 29 | -0.000250 | -0.000250 | -0.000250 | -0.000400 | -0.000250 | -0.000400 | -0.000250 | -0.000250 | -0.000250 |
| 34 | -0.000500 | -0.001100 | -0.001100 | -0.000800 | -0.000800 | -0.000800 | -0.001100 | -0.001100 | -0.000500 |
| 39 | -0.000800 | -0.001250 | -0.001250 | -0.001200 | -0.000140 | -0.001200 | -0.001250 | -0.001250 | -0.000800 |
| 44 | -0.001050 | -0.001400 | -0.001400 | -0.001400 | -0.002000 | -0.001400 | -0.001400 | -0.001400 | -0.001050 |
| 49 | -0.001350 | -0.001600 | -0.001600 | -0.001800 | -0.002200 | -0.001800 | -0.001600 | -0.001600 | -0.001350 |
| 54 | -0.001600 | -0.001800 | -0.001800 | -0.002300 | -0.002150 | -0.002300 | -0.001800 | -0.001800 | -0.001600 |
| 59 | -0.001770 | -0.002000 | -0.002000 | -0.002300 | -0.001900 | -0.002300 | -0.002000 | -0.002000 | -0.001770 |
| 64 | -0.002000 | -0.002250 | -0.002250 | -0.002150 | -0.001800 | -0.002150 | -0.002250 | -0.002250 | -0.002000 |
| 69 | -0.002100 | -0.002450 | -0.002450 | -0.002350 | -0.002370 | -0.002350 | -0.002450 | -0.002450 | -0.002100 |
| 74 | -0.002000 | -0.002600 | -0.002600 | -0.002300 | -0.002150 | -0.002300 | -0.002600 | -0.002600 | -0.002000 |
| 79 | -0.002000 | -0.002750 | -0.002750 | -0.002750 | -0.002500 | -0.002750 | -0.002750 | -0.002750 | -0.002000 |
| 84 | -0.002100 | -0.002860 | -0.002860 | -0.003000 | -0.003400 | -0.003000 | -0.002860 | -0.002860 | -0.002100 |
| 89 | -0.002300 | -0.002960 | -0.002960 | -0.003050 | -0.004800 | -0.003050 | -0.002960 | -0.002960 | -0.002300 |

F-4

CLDR(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | 0.000035 | 0.000035 | 0.000090 | 0.000136 | 0.000184 | 0.000136 | 0.000090 | 0.000035 | 0.000035 |
| 4 | 0.000067 | 0.000067 | 0.000170 | 0.000155 | 0.000183 | 0.000155 | 0.000170 | 0.000067 | 0.000067 |
| 9 | 0.000100 | 0.000100 | 0.000232 | 0.000166 | 0.000176 | 0.000166 | 0.000232 | 0.000100 | 0.000100 |
| 14 | 0.000087 | 0.000087 | 0.000224 | 0.000170 | 0.000157 | 0.000170 | 0.000224 | 0.000087 | 0.000087 |
| 19 | 0.000010 | 0.000010 | 0.000010 | 0.000166 | 0.000800 | 0.000166 | 0.000010 | 0.000010 | 0.000010 |
| 24 | 0.0 | 0.0 | 0.0 | 0.000120 | 0.000033 | 0.000120 | 0.0 | 0.0 | 0.0 |
| 29 | 0.0 | 0.0 | 0.0 | 0.000010 | 0.000003 | 0.000010 | 0.0 | 0.0 | 0.0 |
| 34 | 0.0 | 0.0 | 0.0 | 0.000010 | 0.0 | 0.000010 | 0.0 | 0.0 | 0.0 |
| 39 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 44 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 49 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 54 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 59 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 64 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 69 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 74 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 79 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 84 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 89 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

F-4

CLDA(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -1 | -0.000720 | -0.000750 | -0.000680 | -0.000770 | -0.000760 | -0.000770 | -0.000680 | -0.000750 | -0.000720 |
| 4 | -0.000780 | -0.000790 | -0.000800 | -0.000830 | -0.000800 | -0.000830 | -0.000800 | -0.000790 | -0.000780 |
| 9 | -0.000670 | -0.000720 | -0.000640 | -0.000740 | -0.000730 | -0.000740 | -0.000640 | -0.000720 | -0.000670 |
| 14 | -0.000550 | -0.000620 | -0.000460 | -0.000530 | -0.000520 | -0.000530 | -0.000460 | -0.000620 | -0.000550 |
| 19 | -0.000380 | -0.000500 | -0.000350 | -0.000340 | -0.000360 | -0.000340 | -0.000350 | -0.000500 | -0.000380 |
| 24 | -0.000215 | -0.000350 | -0.000160 | -0.000210 | -0.000230 | -0.000210 | -0.000160 | -0.000350 | -0.000215 |
| 29 | -0.000100 | -0.000200 | -0.000130 | -0.000130 | -0.000150 | -0.000130 | -0.000130 | -0.000200 | -0.000100 |
| 34 | -0.000050 | -0.000110 | -0.000050 | -0.000090 | -0.000099 | -0.000090 | -0.000050 | -0.000110 | -0.000050 |
| 39 | -0.000030 | -0.000060 | -0.000050 | -0.000060 | -0.000060 | -0.000060 | -0.000050 | -0.000060 | -0.000030 |
| 44 | 0.000001 | -0.000030 | -0.000002 | -0.000040 | -0.000030 | -0.000040 | -0.000002 | -0.000030 | 0.000001 |
| 49 | 0.000025 | -0.000010 | -0.000020 | -0.000020 | -0.000010 | -0.000030 | -0.000020 | -0.000010 | 0.000025 |
| 54 | 0.000040 | -0.000010 | 0.0 | -0.000020 | 0.0 | -0.000020 | 0.0 | -0.000010 | 0.000040 |
| 59 | 0.000050 | 0.0 | 0.000010 | -0.000010 | 0.000010 | -0.000010 | 0.000010 | 0.0 | 0.000050 |
| 64 | 0.000070 | 0.0 | 0.000010 | 0.0 | 0.000020 | 0.0 | 0.000010 | 0.0 | 0.000070 |
| 69 | 0.000070 | 0.000010 | 0.000020 | 0.000015 | 0.000020 | 0.000015 | 0.000020 | 0.000010 | 0.000070 |
| 74 | 0.000070 | 0.000010 | 0.000020 | 0.000030 | 0.000020 | 0.000030 | 0.000020 | 0.000010 | 0.000070 |
| 79 | 0.000070 | 0.000020 | 0.000030 | 0.000040 | 0.000020 | 0.000040 | 0.000030 | 0.000020 | 0.000070 |
| 84 | 0.000070 | 0.000010 | 0.000020 | 0.000040 | 0.000020 | 0.000040 | 0.000020 | 0.000010 | 0.000070 |
| 89 | 0.000070 | 0.000020 | 0.000030 | 0.000050 | 0.000020 | 0.000050 | 0.000030 | 0.000020 | 0.000070 |

F111

CFY1(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -10 | 0.547000 | 0.436000 | 0.324000 | 0.145000 | 0.0 | -0.145000 | -0.324000 | -0.436000 | -0.547000 |
| -5 | 0.585000 | 0.464000 | 0.343000 | 0.158000 | 0.0 | -0.158000 | -0.343000 | -0.464000 | -0.585000 |
| 0 | 0.583000 | 0.467000 | 0.351000 | 0.167000 | 0.0 | -0.167000 | -0.351000 | -0.467000 | -0.583000 |
| 5 | 0.572000 | 0.459000 | 0.346000 | 0.173000 | 0.0 | -0.173000 | -0.346000 | -0.459000 | -0.572000 |
| 10 | 0.543000 | 0.443000 | 0.342000 | 0.180000 | 0.0 | -0.180000 | -0.342000 | -0.443000 | -0.543000 |
| 15 | 0.506000 | 0.430000 | 0.353000 | 0.186000 | 0.0 | -0.186000 | -0.353000 | -0.430000 | -0.506000 |
| 20 | 0.471000 | 0.402000 | 0.332000 | 0.189000 | 0.0 | -0.189000 | -0.332000 | -0.402000 | -0.471000 |
| 25 | 0.484000 | 0.390000 | 0.295000 | 0.161000 | 0.0 | -0.161000 | -0.295000 | -0.390000 | -0.484000 |
| 30 | 0.544000 | 0.420000 | 0.295000 | 0.139000 | 0.0 | -0.139000 | -0.295000 | -0.420000 | -0.544000 |
| 35 | 0.618000 | 0.484000 | 0.350000 | 0.172000 | 0.0 | -0.172000 | -0.350000 | -0.484000 | -0.618000 |
| 40 | 0.672000 | 0.541000 | 0.410000 | 0.219000 | 0.0 | -0.219000 | -0.410000 | -0.541000 | -0.672000 |
| 45 | 0.652000 | 0.532000 | 0.412000 | 0.254000 | 0.0 | -0.254000 | -0.412000 | -0.532000 | -0.652000 |
| 50 | 0.612000 | 0.503000 | 0.394000 | 0.239000 | 0.0 | -0.239000 | -0.394000 | -0.503000 | -0.612000 |
| 55 | 0.618000 | 0.496000 | 0.373000 | 0.190000 | 0.0 | -0.190000 | -0.373000 | -0.496000 | -0.618000 |
| 60 | 0.670000 | 0.517000 | 0.363000 | 0.153000 | 0.0 | -0.153000 | -0.363000 | -0.517000 | -0.670000 |
| 65 | 0.715000 | 0.574000 | 0.433000 | 0.120000 | 0.0 | -0.120000 | -0.433000 | -0.574000 | -0.715000 |
| 70 | 0.681000 | 0.588000 | 0.494000 | 0.174000 | 0.0 | -0.174000 | -0.494000 | -0.588000 | -0.681000 |
| 75 | 0.638000 | 0.565000 | 0.492000 | 0.247000 | 0.0 | -0.247000 | -0.492000 | -0.565000 | -0.638000 |
| 80 | 0.619000 | 0.551000 | 0.482000 | 0.289000 | 0.0 | -0.289000 | -0.482000 | -0.551000 | -0.619000 |
| 85 | 0.615000 | 0.546000 | 0.476000 | 0.310000 | 0.0 | -0.310000 | -0.476000 | -0.546000 | -0.615000 |
| 90 | 0.620000 | 0.546000 | 0.471000 | 0.317000 | 0.0 | -0.317000 | -0.471000 | -0.546000 | -0.620000 |



F111

CMX1(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -10 | -0.208800 | -0.118500 | -0.027700 | 0.006800 | 0.0 | -0.006800 | 0.027700 | 0.118500 | 0.208800 |
| -5 | -0.076700 | -0.020900 | 0.041800 | 0.020900 | 0.0 | -0.020900 | -0.041800 | 0.020900 | 0.076700 |
| 0 | 0.167100 | 0.160300 | 0.153500 | 0.055900 | 0.0 | -0.055900 | -0.153500 | -0.160300 | -0.167100 |
| 5 | 0.355500 | 0.320500 | 0.278800 | 0.118500 | 0.0 | -0.118500 | -0.278800 | -0.320500 | -0.355500 |
| 10 | 0.529900 | 0.418200 | 0.306500 | 0.139400 | 0.0 | -0.139400 | -0.306500 | -0.418200 | -0.529900 |
| 15 | 0.669300 | 0.474000 | 0.278800 | 0.139400 | 0.0 | -0.139400 | -0.278800 | -0.474000 | -0.669300 |
| 20 | 0.669300 | 0.453100 | 0.230200 | 0.132600 | 0.0 | -0.132600 | -0.230200 | -0.453100 | -0.669300 |
| 25 | 0.522600 | 0.327300 | 0.132600 | 0.111700 | 0.0 | -0.111700 | -0.132600 | -0.327300 | -0.522600 |
| 30 | 0.452100 | 0.272000 | 0.083500 | 0.048600 | 0.0 | -0.048600 | -0.083500 | -0.272000 | -0.452100 |
| 35 | 0.355500 | 0.230200 | 0.104400 | -0.041800 | 0.0 | 0.041800 | -0.104400 | -0.230200 | -0.355500 |
| 40 | 0.264700 | 0.160300 | 0.055900 | -0.048600 | 0.0 | 0.048600 | -0.055900 | -0.160300 | -0.264700 |
| 45 | 0.418200 | 0.257900 | 0.097600 | -0.035000 | 0.0 | 0.035000 | -0.097600 | -0.257900 | -0.418200 |
| 50 | 0.634300 | 0.453100 | 0.264700 | 0.014100 | 0.0 | -0.014100 | -0.264700 | -0.453100 | -0.634300 |
| 55 | 0.620200 | 0.487600 | 0.348200 | 0.125300 | 0.0 | -0.125300 | -0.348200 | -0.487600 | -0.620200 |
| 60 | 0.606100 | 0.487600 | 0.362300 | 0.195200 | 0.0 | -0.195200 | -0.362300 | -0.487600 | -0.606100 |
| 65 | 0.634300 | 0.501700 | 0.362300 | 0.202000 | 0.0 | -0.202000 | -0.362300 | -0.501700 | -0.634300 |
| 70 | 0.641100 | 0.501700 | 0.362300 | 0.188000 | 0.0 | -0.188000 | -0.362300 | -0.501700 | -0.641100 |
| 75 | 0.647900 | 0.508500 | 0.362300 | 0.181200 | 0.0 | -0.181200 | -0.362300 | -0.508500 | -0.647900 |
| 80 | 0.662000 | 0.501700 | 0.341400 | 0.188000 | 0.0 | -0.188000 | -0.341400 | -0.501700 | -0.662000 |
| 85 | 0.662000 | 0.501700 | 0.341400 | 0.202000 | 0.0 | -0.202000 | -0.341400 | -0.501700 | -0.662000 |
| 90 | 0.669300 | 0.508500 | 0.348200 | 0.222900 | 0.0 | -0.222900 | -0.348200 | -0.508500 | -0.669300 |

F111

CMY1(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -10 | 0.119000 | 0.155000 | 0.191000 | 0.269000 | 0.329000 | 0.269000 | 0.191000 | 0.155000 | 0.119000 |
| -5 | 0.336000 | 0.246000 | 0.155000 | 0.189000 | 0.173000 | 0.189000 | 0.155000 | 0.246000 | 0.336000 |
| 0 | 0.420000 | 0.266000 | 0.113000 | 0.081000 | 0.063000 | 0.081000 | 0.113000 | 0.266000 | 0.420000 |
| 5 | 0.418000 | 0.225000 | 0.032000 | -0.031000 | -0.037000 | -0.031000 | 0.032000 | 0.225000 | 0.418000 |
| 10 | 0.418000 | 0.160000 | -0.098000 | -0.142000 | -0.148000 | -0.142000 | -0.098000 | 0.160000 | 0.418000 |
| 15 | 0.378000 | 0.074000 | -0.229000 | -0.222000 | -0.218000 | -0.222000 | -0.229000 | 0.074000 | 0.378000 |
| 20 | 0.326000 | 0.013000 | -0.299000 | -0.307000 | -0.284000 | -0.307000 | -0.299000 | 0.013000 | 0.326000 |
| 25 | 0.386000 | 0.030000 | -0.326000 | -0.359000 | -0.401000 | -0.359000 | -0.326000 | 0.030000 | 0.386000 |
| 30 | 0.491000 | 0.083000 | -0.324000 | -0.343000 | -0.531000 | -0.343000 | -0.324000 | 0.083000 | 0.491000 |
| 35 | 0.535000 | 0.124000 | -0.287000 | -0.377000 | -0.579000 | -0.377000 | -0.287000 | 0.124000 | 0.535000 |
| 40 | 0.486000 | 0.121000 | -0.244000 | -0.493000 | -0.608000 | -0.493000 | -0.244000 | 0.121000 | 0.486000 |
| 45 | 0.247000 | 0.037000 | -0.174000 | -0.115000 | -0.617000 | -0.115000 | -0.174000 | 0.037000 | 0.247000 |
| 50 | 0.073000 | -0.049000 | -0.170000 | -0.444000 | -0.626000 | -0.444000 | -0.170000 | -0.049000 | 0.073000 |
| 55 | 0.024000 | -0.113000 | -0.249000 | -0.515000 | -0.647000 | -0.515000 | -0.249000 | -0.113000 | 0.024000 |
| 60 | -0.185000 | -0.299000 | -0.412000 | -0.638000 | -0.703000 | -0.638000 | -0.412000 | -0.299000 | -0.185000 |
| 65 | -0.494000 | -0.558000 | -0.621000 | -0.705000 | -0.805000 | -0.705000 | -0.621000 | -0.558000 | -0.494000 |
| 70 | -0.719000 | -0.773000 | -0.826000 | -0.868000 | -0.953000 | -0.868000 | -0.826000 | -0.773000 | -0.719000 |
| 75 | -0.868000 | -0.975000 | -1.082000 | -1.082000 | -1.136000 | -1.082000 | -1.082000 | -0.975000 | -0.868000 |
| 80 | -1.000000 | -1.186000 | -1.371000 | -1.294000 | -1.328000 | -1.294000 | -1.371000 | -1.186000 | -1.000000 |
| 85 | -1.135000 | -1.351000 | -1.567000 | -1.541000 | -1.619000 | -1.541000 | -1.567000 | -1.351000 | -1.135000 |
| 90 | -1.274000 | -1.487000 | -1.700000 | -1.811000 | -1.974000 | -1.811000 | -1.700000 | -1.487000 | -1.274000 |



F111

CMZ1(ALPHA,BETA)

| BETA: | -40 | -30 | -20 | -10 | 0 | 10 | 20 | 30 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| ALPHA | | | | | | | | | |
| -10 | -0.292900 | -0.292900 | -0.292900 | -0.139400 | 0.0 | 0.139400 | 0.292900 | 0.292900 | 0.292900 |
| -5 | -0.292900 | -0.299700 | -0.306500 | -0.146200 | 0.0 | 0.146200 | 0.306500 | 0.299700 | 0.292900 |
| 0 | -0.272000 | -0.285600 | -0.292900 | -0.153500 | 0.0 | 0.153500 | 0.292900 | 0.285600 | 0.272000 |
| 5 | -0.257900 | -0.264700 | -0.272000 | -0.153500 | 0.0 | 0.153500 | 0.272000 | 0.264700 | 0.257900 |
| 10 | -0.243800 | -0.237000 | -0.237000 | -0.146200 | 0.0 | 0.146200 | 0.237000 | 0.237000 | 0.243800 |
| 15 | -0.202000 | -0.257900 | -0.167100 | -0.111700 | 0.0 | 0.111700 | 0.167100 | 0.257900 | 0.202000 |
| 20 | -0.035000 | -0.020900 | -0.006800 | -0.055900 | 0.0 | 0.055900 | 0.006800 | 0.020900 | 0.035000 |
| 25 | 0.216100 | 0.195200 | 0.167100 | 0.048600 | 0.0 | -0.048600 | -0.167100 | -0.195200 | -0.216100 |
| 30 | 0.202000 | 0.222900 | 0.243800 | 0.167100 | 0.0 | -0.167100 | -0.243800 | -0.222900 | -0.202000 |
| 35 | 0.111700 | 0.174400 | 0.237000 | 0.195200 | 0.0 | -0.195200 | -0.237000 | -0.174400 | -0.111700 |
| 40 | 0.243800 | 0.222900 | 0.208800 | 0.174400 | 0.0 | -0.174400 | -0.208800 | -0.222900 | -0.243800 |
| 45 | 0.320500 | 0.230200 | 0.139400 | 0.118500 | 0.0 | -0.118500 | -0.139400 | -0.230200 | -0.320500 |
| 50 | 0.264700 | 0.111700 | -0.035000 | -0.104400 | 0.0 | 0.104400 | 0.035000 | -0.111700 | -0.264700 |
| 55 | 0.237000 | 0.048600 | -0.139400 | -0.272000 | 0.0 | 0.272000 | 0.139400 | -0.048600 | -0.237000 |
| 60 | 0.306500 | 0.083500 | -0.139400 | -0.334600 | 0.0 | 0.334600 | 0.139400 | -0.083500 | -0.306500 |
| 65 | 0.494900 | 0.292900 | 0.090800 | -0.341400 | 0.0 | 0.341400 | -0.090800 | -0.292900 | -0.494900 |
| 70 | 0.501700 | 0.369600 | 0.230200 | -0.160300 | 0.0 | 0.160300 | -0.230200 | -0.369600 | -0.501700 |
| 75 | 0.376400 | 0.264700 | 0.153500 | -0.055900 | 0.0 | 0.055900 | -0.153500 | -0.264700 | -0.376400 |
| 80 | 0.272000 | 0.153500 | 0.035000 | -0.083500 | 0.0 | 0.083500 | -0.035000 | -0.153500 | -0.272000 |
| 85 | 0.257900 | 0.111700 | -0.027700 | -0.097600 | 0.0 | 0.097600 | 0.027700 | -0.111700 | -0.257900 |
| 90 | 0.188000 | 0.062700 | -0.062700 | -0.104400 | 0.0 | 0.104400 | 0.062700 | -0.062700 | -0.188000 |

F111 F(ALPHA ONLY)

| ALPHA | CFX1 | CFZ1 | CMX4 | CMZ4 | CMX2 | CMZ2 | CMX5 | CMX6 | CMZ5 |
|---|---|---|---|---|---|---|---|---|---|
| -10 | -0.009000 | 0.850000 | 0.278762 | -1.184728 | -0.975660 | -0.069694 | -0.005600 | 0.001400 | -0.001400 |
| -5 | -0.025000 | 0.460000 | 0.278762 | -1.184728 | -1.324116 | -0.069694 | -0.005600 | 0.000100 | -0.001400 |
| 0 | -0.030000 | 0.080000 | 0.278762 | -1.184728 | -1.324116 | -0.069694 | -0.005600 | 0.001400 | -0.001400 |
| 5 | -0.028600 | -0.320000 | 0.627218 | -1.184728 | -1.115047 | -0.069694 | -0.005600 | 0.001400 | -0.000700 |
| 10 | 0.009800 | -0.730000 | 0.905980 | -1.254421 | -1.254421 | -0.069694 | -0.006300 | 0.001400 | -0.000700 |
| 15 | 0.045100 | -1.130000 | 1.533183 | -1.533183 | -1.254421 | 0.0 | -0.005600 | 0.000700 | 0.0 |
| 20 | 0.073100 | -1.530000 | 2.160401 | -1.811945 | -1.115047 | 0.069694 | -0.005600 | 0.000700 | 0.000700 |
| 25 | 0.099100 | -1.920000 | 3.345127 | -1.881639 | -1.254421 | 1.324116 | -0.006300 | 0.000700 | 0.000700 |
| 30 | 0.092000 | -2.330000 | 4.460177 | -1.951319 | -1.811945 | 2.508842 | -0.006300 | 0.001400 | 0.001400 |
| 35 | 0.070800 | -2.650000 | 8.084065 | -1.951319 | -2.648232 | 4.042027 | -0.006300 | 0.001400 | 0.002100 |
| 40 | 0.046500 | -1.747000 | 11.080740 | -1.254421 | -3.832958 | 2.787604 | -0.006300 | 0.002100 | 0.002100 |
| 45 | 0.039700 | -1.689600 | 6.899324 | -0.557524 | -4.181412 | 1.811945 | -0.006300 | 0.002800 | 0.002100 |
| 50 | 0.035400 | -1.705400 | 2.439163 | 1.115047 | -3.972347 | 1.324116 | -0.004900 | 0.002800 | 0.002100 |
| 55 | 0.039200 | -1.749200 | 1.742251 | 6.899324 | -3.136060 | 0.975660 | -0.004200 | 0.002100 | 0.002100 |
| 60 | 0.039700 | -1.768000 | 0.905980 | 5.714595 | -1.881639 | -2.160401 | -0.003500 | 0.000700 | 0.002100 |
| 65 | 0.034400 | -1.814200 | 0.209068 | 0.0 | -1.045354 | -3.275435 | -0.003500 | 0.000700 | 0.002100 |
| 70 | 0.035800 | -1.902000 | 0.0 | -0.766592 | -0.696898 | -0.348456 | -0.002800 | 0.0 | 0.002100 |
| 75 | 0.039500 | -1.949000 | 0.069694 | 0.766592 | -0.696898 | -1.045354 | -0.003500 | 0.0 | 0.002100 |
| 80 | 0.040700 | -1.963400 | 0.139374 | 0.766592 | -0.926887 | 0.278762 | -0.004200 | 0.0 | 0.002100 |
| 85 | 0.041200 | -1.969000 | 0.069694 | 0.766592 | -0.975660 | -0.278762 | -0.003500 | -0.000700 | 0.002100 |
| 90 | 0.041200 | -1.969000 | 0.069694 | 0.766592 | -1.045354 | -0.348456 | -0.001400 | -0.001400 | 0.002100 |

F111 F(ALPHA ONLY)

| ALPHA | CFX7 | CFZ7 | CMY7 | CMY3 | CFY6 | CFY5 | CFY2 | CFY4 | CMZ6 |
|---|---|---|---|---|---|---|---|---|---|
| -10 | 0.006100 | -0.012700 | -0.026400 | -26.040000 | 0.003500 | 0.000500 | 0.030000 | 0.120000 | -0.009800 |
| -5 | 0.005500 | -0.015600 | -0.029700 | -26.040000 | 0.003400 | 0.000500 | 0.060000 | 0.130000 | -0.009100 |
| 0 | 0.005000 | -0.010000 | -0.030300 | -26.040000 | 0.003200 | 0.011000 | 0.120000 | 0.160000 | -0.009100 |
| 5 | 0.004300 | -0.014800 | -0.030000 | -24.420000 | 0.003100 | 0.000600 | 0.190000 | 0.130000 | -0.008400 |
| 10 | 0.003500 | -0.014300 | -0.030500 | -22.790000 | 0.002900 | 0.000600 | 0.230000 | 0.010000 | -0.009100 |
| 15 | 0.002600 | -0.015400 | -0.031200 | -26.230000 | 0.003200 | 0.001000 | 0.240000 | 0.0 | -0.009100 |
| 20 | 0.001700 | -0.018400 | -0.032000 | -29.670000 | 0.003200 | 0.001300 | 0.230000 | 0.430000 | -0.009100 |
| 25 | 0.000800 | -0.020700 | -0.034400 | -33.700000 | 0.003300 | 0.001800 | 0.260000 | 1.050000 | -0.009100 |
| 30 | -0.000300 | -0.024100 | -0.037000 | -37.720000 | 0.003200 | 0.001400 | 0.290000 | 1.200000 | -0.009100 |
| 35 | -0.001200 | -0.025000 | -0.036100 | -42.900000 | 0.002900 | 0.001300 | 0.290000 | 0.790000 | -0.008400 |
| 40 | -0.002100 | -0.022000 | -0.031600 | -44.690000 | 0.002500 | 0.002700 | 0.560000 | 0.230000 | -0.006300 |
| 45 | -0.003500 | -0.016900 | -0.024700 | -41.810000 | 0.001900 | 0.001400 | 1.230000 | -0.180000 | -0.004900 |
| 50 | -0.004300 | -0.014400 | -0.017400 | -19.000000 | 0.001900 | 0.000700 | 1.700000 | 0.780000 | -0.004200 |
| 55 | -0.004800 | -0.012000 | -0.011400 | 2.000000 | 0.004100 | 0.000200 | 1.540000 | 2.610000 | -0.004200 |
| 60 | -0.004500 | -0.010300 | -0.008700 | -7.000000 | 0.001600 | -0.001300 | -0.140000 | 2.270000 | -0.004200 |
| 65 | -0.004700 | -0.009300 | -0.003400 | -30.000000 | -0.000500 | -0.003500 | -1.180000 | 0.490000 | -0.004200 |
| 70 | -0.005200 | -0.010500 | -0.004000 | -27.000000 | 0.000400 | -0.003500 | -0.090000 | -0.180000 | -0.004200 |
| 75 | -0.005300 | -0.010600 | -0.004000 | -4.000000 | 0.000500 | -0.002700 | 0.640000 | -0.010000 | -0.004200 |
| 80 | -0.005200 | -0.008200 | -0.004000 | -3.000000 | 0.0 | -0.002500 | 0.580000 | 0.090000 | -0.004200 |
| 85 | -0.005600 | -0.007100 | -0.004000 | -17.000000 | -0.000500 | -0.002400 | 0.610000 | 0.100000 | -0.004200 |
| 90 | -0.061000 | -0.007700 | -0.004000 | -24.000000 | -0.000800 | -0.002400 | 0.730000 | 0.160000 | -0.004200 |

F-4 F(ALPHA ONLY)

| ALPHA | CYP | CLR | CLP | CNR | CNP | CMQ | CYR |
|---|---|---|---|---|---|---|---|
| -1 | 0.193000 | 0.024120 | -0.506500 | -1.061000 | 0.0 | -4.700000 | 2.050000 |
| 4 | 0.699500 | 0.108500 | -0.651300 | -0.844200 | -0.084420 | -4.800000 | 1.881000 |
| 9 | 0.482400 | 0.241200 | -0.639200 | -0.916600 | -0.048240 | -5.200000 | 1.761000 |
| 14 | 0.554800 | 0.482400 | -0.542700 | -0.771900 | -0.024120 | -6.200000 | 1.423000 |
| 19 | 0.916600 | 1.085000 | -0.603000 | -1.206000 | 0.048240 | -6.100000 | 1.544000 |
| 24 | 1.471000 | 1.713000 | -0.964800 | -1.857000 | 0.072360 | -7.100000 | 1.110000 |
| 29 | 1.930000 | 1.375000 | -0.964800 | -1.206000 | 0.048240 | -6.900000 | -0.964800 |
| 34 | 1.254000 | 0.964800 | -0.880400 | -0.458300 | -0.120600 | -9.200000 | -2.653000 |
| 39 | -0.048240 | 0.603000 | -0.482400 | -0.193000 | -0.313600 | -10.000000 | -2.412000 |
| 44 | -1.134000 | 0.325600 | -0.337700 | -0.217100 | -0.337700 | -11.200000 | -1.978000 |
| 49 | -0.651300 | 0.120600 | -0.289400 | -0.482400 | -0.144700 | -10.500000 | -1.930000 |
| 54 | -1.688000 | 0.036180 | -0.241200 | -0.940700 | -0.385900 | -8.800000 | -0.964800 |
| 59 | -2.002000 | 0.048240 | -0.313600 | -0.964800 | -0.554800 | -7.200000 | -2.388000 |
| 64 | -1.182000 | 0.096480 | -0.410000 | -0.868300 | -0.361800 | -7.100000 | -2.002000 |
| 69 | -1.688000 | 0.193000 | -0.518600 | -0.578900 | -0.120600 | -6.700000 | -1.254000 |
| 74 | -1.110000 | 0.193000 | -0.639200 | -0.120600 | 0.072360 | -7.100000 | -0.699500 |
| 79 | -1.013000 | 0.108500 | -0.735700 | 0.434200 | 0.337700 | -6.600000 | -0.385900 |
| 84 | -1.664000 | 0.024120 | -0.844200 | 0.603000 | 0.554800 | -6.500000 | -0.506500 |
| 89 | -1.930000 | 0.0 | -0.916600 | 0.603000 | 0.892500 | -6.200000 | -0.892500 |

THE BELOW LISTED COEFFICIENTS HAVE BEEN EQUATED TO ZERO

| | | | | | | |
|---|---|---|---|---|---|---|
| CYDA | CXR | CXDR | CXP | CXQ | CXDA | CYQ |
| CYDH | CZP | CZQ | CZE | CZDA | CZDR | CLQ |
| CLDS | CMP | CMR | CMDA | CMDR | CNQ | CNDH |

# BIBLIOGRAPHY


1. Buehler, R. R., The Derivation, Solution, and Analysis of Airplane Spin Equations Modeled in an Inertial Coordinate System, Master's Thesis, United States Naval Postgraduate School, Monterey, California, March 1972.

2. Grumman Aircraft Engineering Corporation Report No. 394-68-1, Volume I, The Spin Behavior of Aircraft, by William Bihrle Jr., and Arthur C. Heyman, 1967.

3. National Aeronautics and Space Administration TN D-6117, Utilization of a Fixed Base Simulator to Study the Stall and Spin Characteristics of Fighter Airplanes, by B. L. Moore, E. L. Anglin, M. S. Adams, P. L. Deal, and L. H. Person, Jr., March 1971.

4. McDonnell Douglas Aircraft Corp. Report MDC A0005, Vol I, Model F-4 Spin Evaluation Program, by C. C. Brady, W. A. Moran, and M. L. Rosenstein, 1969.

5. Wells, D. A., "Theory and Problems of Lagrangian Dynamics", Schaum's Outline Series, McGraw-Hill Book Company, 1967.

6. National Advisory Committee for Aeronautics TN 2181, The Aerodynamic Forces and Moments on a 1/10 Scale Model of a Fighter Airplane in Spinning Attitudes as Measured on a Rotary Balance in the Langley 20-Foot Free Spinning Tunnel, by R. W. Stone, Jr., M. Burk, Jr., and W. Bihrle, Jr., 1950.

7. Johnson, A. E., A Literature Survey of the Problem of Aircraft Spins, Master's Thesis, United States Naval Postgraduate School, Monterey, California, September 1971.

8. Beckey, G. A., Karplus, W. J., Hybrid Computation, John Wiley and Sons, Inc., 1968.

## INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 2 |
| 2. | National Aeronautics and Space Administration<br>Langley Research Center<br>Hampton, Virginia 23365 | 1 |
| 3. | Library, Code 0212<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| 4. | Chairman, Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 5. | Professor L. V. Schmidt, Code 57Sx<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 6. | Professor T. H. Gawain, Code 57<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 7. | Professor M. H. Redlin, LTJG, USNR, Code 57Rp<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 8. | Lieutenant R. L. Champoux, USN<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

| | |
|---|---|
| INATING ACTIVITY (Corporate author)<br>Naval Postgraduate School<br>Monterey, California 93940 | 2a. REPORT SECURITY CLASSIFICATION<br>Unclassified<br>2b. GROUP |

ORT TITLE

The Numerical Solution and Analysis of Airplane Spin Equations Modeled in a Fixed Coordinate System

CRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1972

THOR(S) (First name, middle initial, last name)

Robert Louis Champoux

| | | |
|---|---|---|
| ORT DATE<br>December 1972 | 7a. TOTAL NO. OF PAGES<br>140 | 7b. NO. OF REFS<br>8 |
| NTRACT OR GRANT NO.<br>ROJECT NO. | 9a. ORIGINATOR'S REPORT NUMBER(S) | |
| | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) | |

STRIBUTION STATEMENT

Approved for public release; distribution unlimited

| | |
|---|---|
| UPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY<br>Naval Postgraduate School<br>Monterey, California 93940 |

BSTRACT

Three forms of the airplane spin equations of motion, derived by Buehler in Reference [1], form the basis for the development of a computer program designed to seek dynamically stable equilibrium solutions of a spinning aircraft. The program incorporates two solution techniques: one based upon Euler integration, the other, a version of minimization by gradient search. Secondary programs are developed to (1) generate power off glide parameters for use in the validation of the equations of motion, and (2) evaluate equation residuals obtained from a grid of initial conditions over the potential solution space. F-111 and F-4 aerodynamic force and moment models were utilized to evaluate the solution methods and equations of motion. The numerical results indicate that the F-111 and F-4 data are not representative of the actual aircraft and, therefore, it is highly unlikely that dynamically stable equilibrium solutions can be achieved from these models. The utility of the two solution methods is evaluated and the numerical results are analyzed in order to gain insight into the optimal application of the three forms of the equations of motion. The paper concludes with a discussion concerning the qualitative validation of the equations of motion.

D FORM 1 NOV 68 1473 (PAGE 1)

0101-807-6811



Security Classification

A-31408

| KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Airplane Spin Equations | | | | | | |
| F-4 Spin | | | | | | |
| F-111 Spin | | | | | | |
| Gradient Search Solution Method | | | | | | |

141253
Thesis
C365
c.1
Champoux
The numerical solution and analysis of airplane spin equations modeled in a fixed coordinate system.

Thesis
C365
c.1
141253
Champoux
The numerical solution and analysis of airplane spin equations modeled in a fixed coordinate system.

thesC365
The numerical solution and analysis of a
3 2768 002 09705 7
DUDLEY KNOX LIBRARY